Simulink® Control Design

For Use with Simulink®

- Modeling
- Simulation
- Implementation

Getting Started

Version 1

The MathWorks

Getting Started with Simulink Control Design

Printing History: June 2004 Online only New for Version 1.0 (Release 14)

Contents

3 Linearizing Models

4 Linearizing Models Using Functions

Introduction

What Is Simulink Control Design?

Simulink® Control Design provides tools for linearization of control systems and physical models in Simulink. Linearized Simulink models often simplify system analysis and compensator design. This is useful in many industries and applications, including

- Aerospace: flight control, guidance, navigation
- Automotive: cruise control, emissions control, transmission
- Equipment manufacturing: motors, disk drives, servos

Simulink Control Design builds on features of the Control System Toolbox, such as LTI Viewer and SISO Design Tool, as well as the Simulink linearization engine. Use Simulink Control Design to

- Specify input and output points without inserting blocks in the model
- Perform open loop analysis without deleting feedback loops
- Extract operating points from a simulation
- Compute operating points from implicitly defined state and output values (also known as trimming)
- Inspect and debug results of the linearization
- Export results to the MATLAB® workspace
- Manage linearization projects
- Use new linearization and trim functions

Using the GUI vs. Command Line Functions

The Simulink Control Design GUI provides a graphical environment for control system linearization and design. Using it, you can easily inspect and analyze operating points and results of linearization. In addition, you can save and restore settings as well as export results to the MATLAB workspace.

You can also linearize models using the command line functions. With the functions, you can automate many of your linearization projects and perform *batch linearization*, such as linearization of a system at several different values of a parameter.

Using the Documentation

Expected Background

Users of this guide should be familiar with control systems design and analysis, and have experience creating Simulink models. Familiarity with the Control System Toolbox is also desirable.

How to Use This Guide

If you are new to linearization, begin with Chapter 2, "What Is Linearization?." This introduces linearization concepts that are important for accurate creation and use of linearized models.

All users should read Chapter 3, "Linearizing Models," which describes features and use of the Simulink Control Design GUI.

To automate the linearization process, or perform batch linearization, continue with Chapter 4, "Linearizing Models Using Functions."

Online Documentation

Further documentation is available online or in the Help browser, including the following chapters: "Understanding and Controlling Results," "Function Reference," and "Block Reference."

Using Examples and Demos

The Simulink Control Design documentation makes use of several examples. You can access these examples by typing

```
demo
```

at the MATLAB prompt and selecting **Simulink Control Design** under the **Simulink** node. You can also access the examples by clicking the **Demos** tab in the Help browser.

Related Products

The MathWorks provides several products that are especially relevant to the kinds of tasks you can perform with Simulink Control Design. For more information about these products, visit the MathWorks Web site, at `http://www.mathworks.com/products/simcontrol/`.

What Is Linearization?

Engineers often use linearization in the design and analysis of control systems and physical models. Successful linearization depends on an understanding of important linearization concepts and factors affecting linearization, discussed here.

Purpose of Linearization

Many common control system analysis and design methodologies require linear, time-invariant models. However, control systems and physical models created with Simulink are often nonlinear and time-varying.

Linearization is the approximation of a nonlinear system as a linear system, based on the assumption that the system is almost linear within a certain range of operation. With a linearized model you can

- Use the Control System Toolbox LTI Viewer to display and analyze dynamic behaviors of a model
- Tune feedback gains and design compensators for a system
- Express a model as a transfer function, state space model, or zero-pole-gain model
- Determine the response of a model to arbitrary input signals

A linearized model can provide a good approximation to a nonlinear system when created and used carefully. Factors affecting the accuracy of the approximation addressed in this chapter are

- Choice of operating points. See “Operating Points” on page 2-3.
- Understanding the equations for the linearized model. See “Linearization of Nonlinear Models” on page 2-5.
- Controlling the effect of feedback loops. See “Understanding Open Loop Analysis” on page 2-11.

Understanding Linearization

Understanding the process of linearization with Simulink Control Design depends on first understanding several concepts such as operating points (define the point at which the linearization takes place), the relationship between linearization theory and linearization of Simulink models, and the effects of feedback loops. The remainder of this chapter discusses these concepts in the following sections.

Subsequent chapters give information on linearization using Simulink Control Design.

Operating Points

The operating point of a dynamic system defines its overall *state* at a given time. For example, in a model of a car engine, variables such as engine speed, throttle angle, engine temperature, and surrounding atmospheric conditions typically describe the operating point. The level of the operating point affects the system’s behavior. For example, the behavior of a car engine can vary greatly when it operates at high or low elevations.

A linearized model is an approximation that is valid in a small region around the operating point of the system. Near the operating point the approximation will be good, while far away it will be poor. A linearized model of a car being operated at 3000 ft. will be very accurate at elevations close to 3000 ft. but less accurate as the car travels higher or lower.

The following figure shows a nonlinear function, $y = x^2$, and a linear function, $y = 2x - 1$. The linear function is an approximation to the nonlinear function about the operating point x=1, y=1. Near this operating point, the approximation is good. Away from this operating point, the approximation is poor. The precise boundaries of this region are often somewhat arbitrary. The following figure shows a possible region of good approximation for the linearization of $y = x^2$.

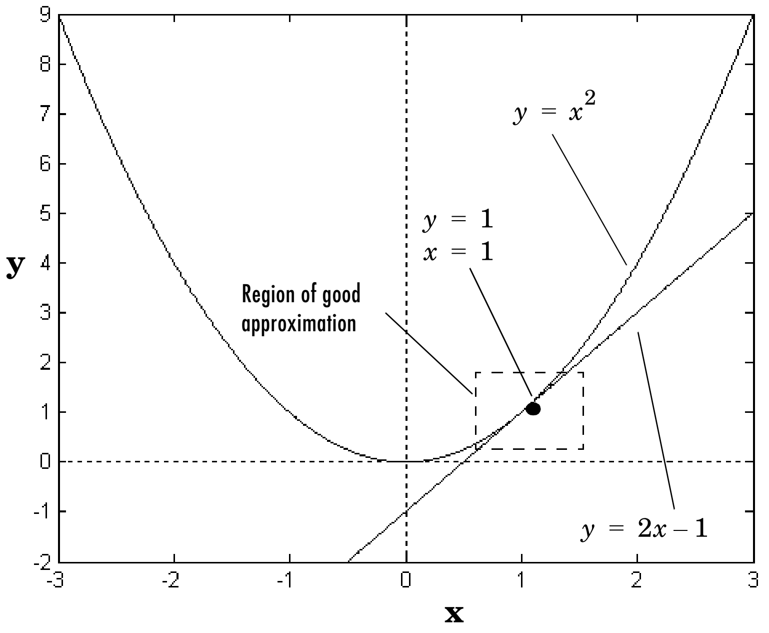


Figure 2-1: Linear Approximation

When creating a linearized model from a Simulink model, the choice of operating point is important as it will determine the accuracy of the approximation. Choose an operating point that is very close to the expected operating values of the system. One way to do this is with an equilibrium operating point, described in the next section.

Equilibrium Operating Points

The equilibrium operating point remains steady and constant with time. It is also known as a steady state or trimmed operating point. For example, a car operating on cruise control on a flat road maintains a constant speed. Its operating point is steady, or at equilibrium, although a controller stabilizes the system.

A hanging pendulum provides an example of a stable equilibrium operating point. When the pendulum hangs straight down it's position does not change with time since it is at an equilibrium position. When it's position deviates slightly from this position, it always returns to the equilibrium; small changes in the operating point do not cause the system to leave the region of good approximation around the equilibrium value.

An example of an unstable equilibrium operating point is a pendulum that points upwards. As long as the pendulum points *exactly* upwards, it is steady at this equilibrium state. However, when the pendulum deviates slightly from this state, it swings downwards and the operating point leaves the region around the equilibrium value.

A model linearized about a stable equilibrium operating point is likely to remain within the region around the equilibrium value. This model will give a good approximation to the nonlinear model. A model linearized about an unstable equilibrium operating point is not likely to remain within the region around the equilibrium value. This model gives a poor approximation to the nonlinear model once it deviates from this equilibrium point, unless a controller is designed to stabilize the system. The magnetic ball system in Chapter 3, "Linearizing Models" gives an example of an unstable equilibrium that is stabilized by a controller.

Simulink Control Design provides several methods for specifying the operating point of a model. "Specifying Operating Points" on page 3-18, and "Specifying Operating Points Using Functions" on page 4-12 describe these methods.

Linearization of Nonlinear Models

The following figure shows a block diagram consisting of an external input signal, $u(t)$, a measured output signal, $y(t)$, and a nonlinear system that describes the system's states and its dynamic behavior, P.

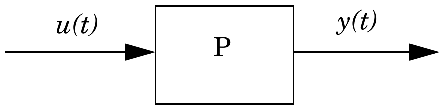


Figure 2-2: Block Diagram

The block diagram is a visual representation of a nonlinear system. You can also express the system in terms of the state space equations

$$\begin{aligned}\dot{x}(t) &= f(x(t), u(t), t)\\ y(t) &= g(x(t), u(t), t)\end{aligned}$$

where $x(t)$ represents the system's states, $u(t)$ represents the inputs, and $y(t)$ represents the outputs. In these equations, the variables vary continuously with time. Discrete-time and multi-rate models are discussed in the section "Linearization of Discrete-Time Models" on page 2-7 and "Linearization of Multi-Rate Models" on page 2-8.

A linear time-invariant approximation to this system is valid in a region around the operating point at $t=t_0$, $x(t_0)=x_0$, and $u(t_0)=u_0$. If the values of the system's states, $x(t)$ and inputs, $u(t)$ are close enough to the operating point, the system will behave approximately linearly.

To describe the linearized model, it helps to first define a new set of variables centered about the operating point of the states, inputs, and outputs:

$$\begin{aligned}\delta x(t) &= x(t) - x_0\\ \delta u(t) &= u(t) - u_0\\ \delta y(t) &= y(t) - y_0\end{aligned}$$

The value of the outputs at the operating point is given by $y(t_0)=g(x_0,u_0,t_0)=y_0$.

Note When comparing a linearized model with the original model, remember that the convention used in this book is to write the linearized model in terms of δx, δu, and δy. The value of each of these variables at the operating point is zero.

The linearized state space equations written in terms of $\delta x(t)$, $\delta u(t)$, and $\delta y(t)$ are

$$\begin{aligned}\delta\dot{x}(t) &= A\delta x(t) + B\delta u(t)\\ \delta y(t) &= C\delta x(t) + D\delta u(t)\end{aligned}$$

where A, B, C, and D are constant coefficient matrices. These matrices are defined as the Jacobians of the system, evaluated at the operating point

$$A = \left.\frac{\partial f}{\partial x}\right|_{t_0,x_0,u_0} \qquad B = \left.\frac{\partial f}{\partial u}\right|_{t_0,x_0,u_0}$$

$$C = \left.\frac{\partial g}{\partial x}\right|_{t_0,x_0,u_0} \qquad D = \left.\frac{\partial g}{\partial u}\right|_{t_0,x_0,u_0}$$

The transfer function of the linearized model can be used in place of the system, P, in the diagram of Figure 2-2. To find the transfer function, divide the Laplace transform of $\delta y(t)$ by the Laplace transform of $\delta u(t)$:

$$P_{lin}(s) = \frac{\delta Y(s)}{\delta U(s)}$$

Linearization of Discrete-Time Models

Discrete-time models are similar to continuous models, discussed in the previous section, with the exception that the values of system variables change at discrete times, t_k, where k is an integer value. The state-space equations for a nonlinear, discrete-time system are

$$\begin{aligned}x_{k+1} &= f(x_k, u_k, t_k)\\ y_k &= g(x_k, u_k, t_k)\end{aligned}$$

A linear time-invariant approximation to this system is valid in a region around the operating point

$$t_k = t_{k_0}, x_k = x_{k_0}, u_k = u_{k_0}, \text{ and } y_k = g(x_{k_0}, u_{k_0}, t_{k_0}) = y_{k_0}$$

If the values of the system's states, x_k, inputs, u_k, and outputs, y_k, are close enough to the operating point, the system will behave approximately linearly. As with continuous time systems it is helpful to define variables centered about the operating point values

$$\begin{aligned} \delta x_k &= x_k - x_{k_0} \\ \delta u_k &= u_k - u_{k_0} \\ \delta y_k &= y_k - y_{k_0} \end{aligned}$$

where the value of the outputs at the operating point are defined as:

$$y_{k_0} = g(x_{k_0}, u_{k_0}, t_{k_0})$$

The linearized state-space equations can then be written in terms of these new variables

$$\begin{aligned} \delta x_{k+1} &\approx A\delta x_k + B\delta u_k \\ \delta y_k &\approx C\delta x_k + D\delta u_k \end{aligned}$$

where A, B, C, and D are given by

$$A = \left.\frac{\partial f}{\partial x_k}\right|_{t_0, x_0, u_0} \qquad B = \left.\frac{\partial f}{\partial u_k}\right|_{t_0, x_0, u_0}$$

$$C = \left.\frac{\partial g}{\partial x_k}\right|_{t_0, x_0, u_0} \qquad D = \left.\frac{\partial g}{\partial u_k}\right|_{t_0, x_0, u_0}$$

Linearization of Multi-Rate Models

Multi-rate models involve states with various sampling rates. This means that the state variables change values at different times and with different frequencies, with some variables possibly changing continuously. The general state-space equations for a nonlinear, multi-rate system are

$$\begin{aligned}
\dot{x}(t) &= f\left(x(t), x_1(k_1), \ldots, x_m(k_m), u(t), t\right) \\
x_1(k_1+1) &= f_1\left(x(t), x_1(k_1), \ldots, x_m(k_m), u(t), t\right) \\
\vdots & \qquad\qquad \vdots \\
x_m(k_m+1) &= f_i\left(x(t), x_1(k_1), \ldots, x_m(k_m), u(t), t\right) \\
y(t) &= g\left(x(t), x_1(k_1), \ldots, x_m(k_m), u(t), t\right)
\end{aligned}$$

where k_1,..., k_m are integer values and t_{k_1} ,..., t_{k_m} are discrete times. The linearized equations will approximate tnis system as a single-rate discrete model:

$$\begin{aligned}
\delta x_{k+1} &\approx A\delta x_k + B\delta u_k \\
\delta y_k &\approx C\delta x_k + D\delta u_k
\end{aligned}$$

Linearization of Simulink Models

You apply the linearization concepts when linearizing Simulink models with Simulink Control design. Simulink uses a series of connected *blocks* to model physical systems and control systems. Input and output signals connect the blocks, which represent mathematical operations. The nonlinear system, P, in Figure 2-2, now represents a series of connected Simulink blocks.

Simulink linearizes both continuous and discrete-time nonlinear systems by computing the state-space matrices, A, B, C, and D, using one of the linearization algorithms described in "Understanding and Controlling Results" in the online documentation. A common method is to perturb the inputs and states by a small amount about their operating point values. The response to these perturbations is measured in the outputs. Both the perturbations and the response are used to compute the matrices. Refer to "Understanding and Controlling Results" in the online documentation for more information.

To linearize models with Simulink Control Design, you must insert linearization input and output points in the model diagram. A linearization input point defines an input to the linearized model while a linearization output point defines an output of the linearized model. Additionally, when the linearized models are computed using numerical perturbation, an input point is the point on the diagram where the small perturbation to the input signal is introduced and an output point is the point on the diagram where the response to this perturbation is measured. “Selecting Linearization Points” on page 3-10 gives methods for selecting linearization input and output points with Simulink Control Design.

Understanding Open Loop Analysis

Many control systems contain feedback loops. An example of such a system is shown in this figure.

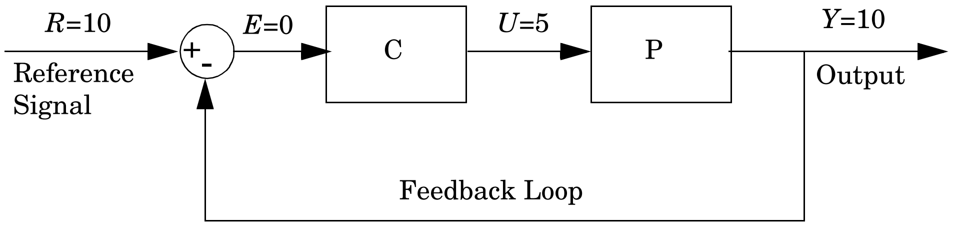


Figure 2-3: Control System with Feedback Loop

The model represented in this figure is at equilibrium. Consider linearizing the plant, P, about this equilibrium operating point by changing the input signal, *U*, by a small amount, *u*, and measuring the change in the output signal, *y*. The portion of the system that you want to linearize is shown in the following figure.

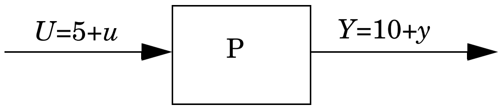


Figure 2-4: Portion of System for Linearization

However, due to the presence of the feedback loop, the change in the output signal will feed back into the controller, C, and then into the plant. This affects the behavior of the system you are linearizing. In fact, if C and P were linear, the linearized model between U and Y would be $\frac{P(s)}{1+C(s)P(s)}$ rather than $P(s)$.

You could manually remove the feedback signal from the model in an attempt to resolve this issue. However, as shown in the following figure, this changes the operating point of the system since the error signal, E, is now equal to the reference signal, R. Linearizing about this new operating point would change the linearization results. Of course, this only makes a difference for non-linear models. When the model is already linear, it has the same form regardless of the operating point.

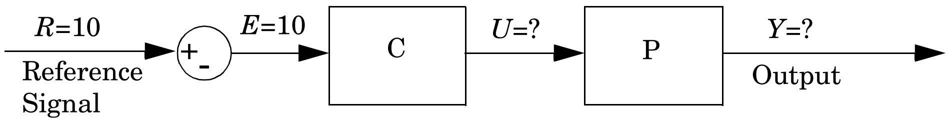


Figure 2-5: Control System with Feedback Signal Removed

When linearizing Simulink models, use Simulink Control Design's capability to label an input or output point as open loop. Doing so ensures that the output signal is not fed back into the model but keeps the operating point the same. In other words, in the linear case, you would compute $P(s)$ rather than $\frac{P(s)}{1+C(s)P(s)}$. "Performing Open Loop Analysis" on page 3-13 and "Open Loop Analysis Using Functions" on page 4-11 give methods for assigning open loop points in Simulink models using the Simulink Control Design GUI and functions respectively.

Linearizing Models

Use the Simulink Control Design Graphical User Interface (GUI) to linearize Simulink models. With this tool you can insert, inspect, and change input and output points for linearization without adding blocks to the model diagram, perform open loop analysis without manually breaking signal lines, and specify or compute operating points for linearization. Additionally, use the GUI to inspect operating points and results of the linearization (block-by-block), export them to the MATLAB workspace, and display them in the LTI Viewer. You can also save, reload, and compare linearization projects using alternative operating points and parameter values.

Overview

The main steps to linearize a model using the Simulink Control Design GUI are

1. Creating or opening a model
2. Opening a linearization project in the Control and Estimation Tools Manager
3. Configuring the model
4. Specifying the operating point
5. Linearizing the model or block
6. Analyzing the results
7. Exporting and saving your work

The following section introduces an example containing a nonlinear system (a levitating magnetic ball). Subsequent sections of this chapter use the example for a detailed discussion of each step.

You can also use the Simulink Control Design command line functions to linearize a model. Chapter 4, "Linearizing Models Using Functions" gives detailed information on these functions. For a discussion of the advantages and disadvantages of the GUI versus the command line interface, please refer to "Using the GUI vs. Command Line Functions" on page 1-2.

Example: The Magnetic Ball

The electronic circuit in the following figure consists of a voltage source, a resistor, and an inductor in the form of a tightly wound coil. An iron ball beneath the inductor experiences a gravitational force as well as an induced magnetic force (from the inductor) that opposes the gravitational force.

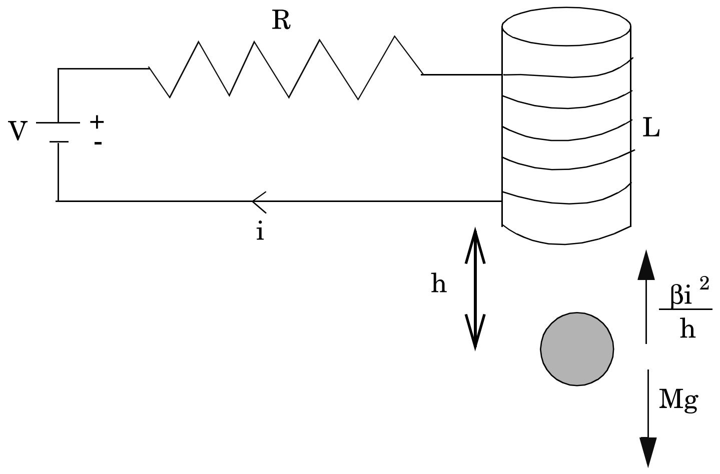


Figure 3-1: Schematic Diagram of Magnetic Ball System

Model Equations

A differential equation for the force balance on the ball is given by

$$M\frac{d^2h}{dt^2} = Mg - \frac{\beta i^2}{h}$$

where M is the mass of the ball, h is the height (position) of the ball, g is the acceleration due to gravity, i is the current, and β is a constant related to the magnetic force experienced by the ball. This equation describes the height, h, of the ball due to the unbalanced forces acting upon it.

The current in the circuit also varies with time and is given by the following differential equation

$$L\frac{di}{dt} = V - iR$$

where L is the inductance of the coil, V is the voltage in the circuit, and R is the resistance of the circuit.

The system of equations has three states

$$h, \frac{dh}{dt}, i$$

The system also has one input (V), and one output (h). It is a nonlinear system due to the term in the equation involving the square of i and the inverse of h. Due to its nonlinearity, you cannot analyze this system using methods for linear-time-invariant (LTI) systems such as step response plots, bode diagrams, and root-locus plots. However, linearizing using Simulink Control Design approximates the nonlinear system as an LTI system. This linearized system can then use the LTI Viewer for display and analysis. Refer to “Purpose of Linearization” on page 2-2 for a discussion of the uses of linearized models and “Linearization of Nonlinear Models” on page 2-5 for a discussion of the linearization process.

Creating or Opening a Simulink Model

The first step in the linearization process is to create or open a Simulink model of your system. The model can have any number of inputs and outputs (including none), and any number of states. The model can include user-defined blocks or S-functions. Your model can involve a plant only, a plant with a feedback loop and controller, or any number of subsystems.

To open the model for the magnetic ball example, type

```
magball
```

at the MATLAB prompt. The magnetic ball system opens in Simulink as shown in this figure.

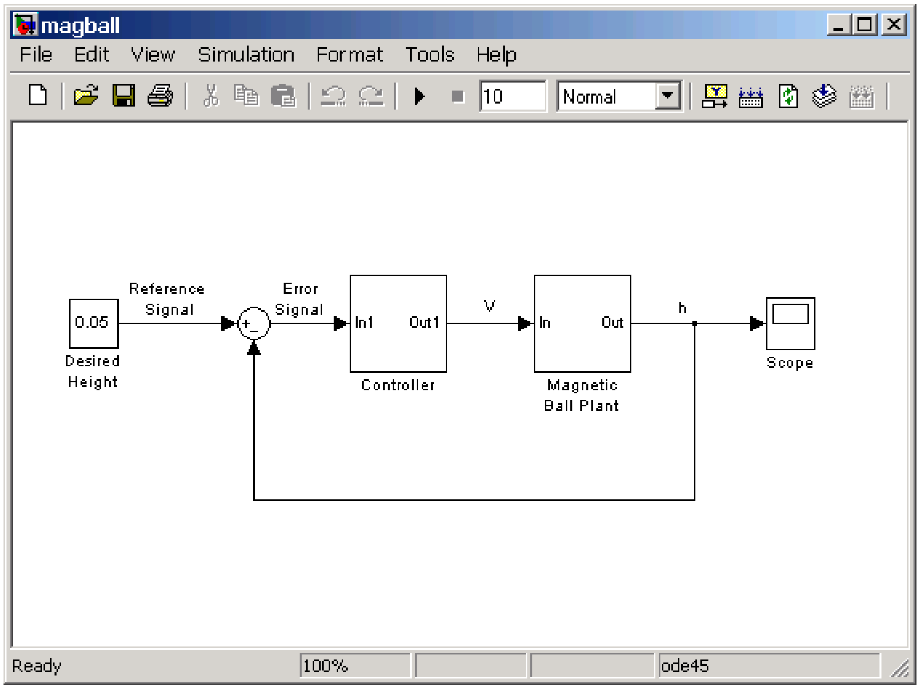


Figure 3-2: Simulink Model of Magnetic Ball System

The `magball` model consists of

- The magnetic ball system itself, within the subsystem labeled Magnetic Ball Plant.
- A Controller subsystem that controls the height of the ball by balancing the forces acting on it.
- A reference signal that sets the desired height of the ball.
- A Scope block that displays the height of the ball as a function of time.

Double-click a block to view its contents. The Controller block contains a zero-pole-gain model. The Magnetic Ball Plant block is shown in this figure.

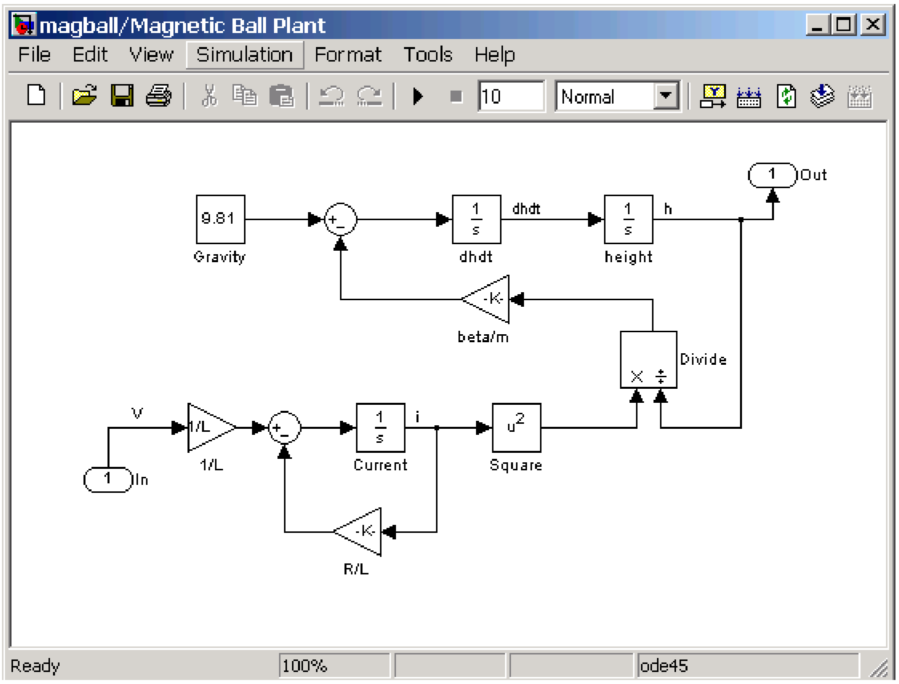


Figure 3-3: Plant Subsystem of the Magnetic Ball Model

The input to the Magnetic Ball Plant system, which is also the output of the Controller subsystem, is the voltage, *V*. The output is the height of the ball, *h*.

The system contains three states within the three integrators: `height`, `dhdt`, and `Current`.

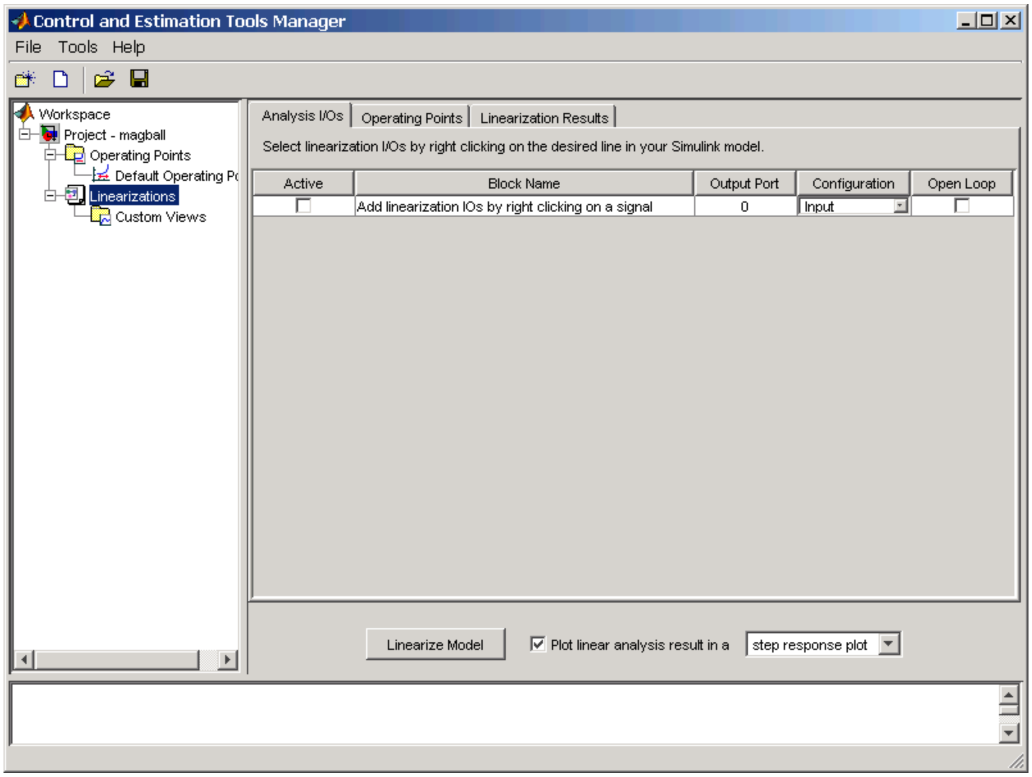
Values of the parameters are given as: M=0.1 kg, g=9.81 m/s^2, R=2 Ohm, L=0.02 H, β=0.001.

Beginning a New Linearization Project

The Control and Estimation Tools Manager provides a graphical environment for performing, and managing, control and estimation tasks such as linearization. Within this environment you can create projects which can include several tasks.

Open a new project for linearization by choosing **Tools -> Control Design -> Linear Analysis** from the magnetic ball model.

The Control and Estimation Tools Manager opens, as shown in the following figure.



Select linearization I/Os by right clicking on the desired line in your Simulink model.

Active	Block Name	Output Port	Configuration	Open Loop
☐	Add linearization IOs by right clicking on a signal	0	Input	☐

☑ Plot linear analysis result in a step response plot

Figure 3-4: Control and Estimation Tools Manager Window

The Control and Estimation Tools Manager creates and manages projects that use tools such as Simulink Control Design. The left panel is the project tree, which contains all your current projects. At this stage you should have just one project, **Project - magball**. Select a node within the tree to display its contents in the panel on the right.

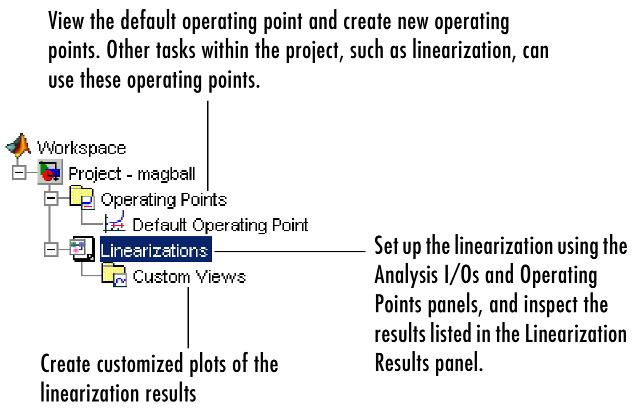

View the default operating point and create new operating points. Other tasks within the project, such as linearization, can use these operating points.
Workspace
Project - magball
Operating Points
Default Operating Point
Linearizations
Custom Views
Set up the linearization using the Analysis I/Os and Operating Points panels, and inspect the results listed in the Linearization Results panel.
Create customized plots of the linearization results

Figure 3-5: Project Tree

- For information on the **Operating Points** node or the **Operating Points** panel within the **Linearizations** node, refer to “Specifying Operating Points” on page 3-18.
- For information on the **Analysis I/Os** panel within the **Linearizations** node, refer to “Configuring the Model for Linearization” on page 3-10.
- For information on the **Linearization Results** panel within the **Linearizations** node and inspecting linearization results, refer to “Analyzing the Results” on page 3-34.
- For information on **Custom Views**, refer to “Analyzing the Results” on page 3-34.

Configuring the Model for Linearization

Before linearizing the model, you must first configure the model diagram. This involves selecting input and output linearization points, and then, if necessary, inserting open loop points for performing open loop analysis. Inspect the selected linearization points using the **Analysis I/Os** panel within the **Linearizations** node of the Control and Estimation Tools Manager.

A linearization input point defines an input to the linearized model while a linearization output point defines an output of the linearized model. Additionally, when the linearized models are computed using numerical perturbation, an input point is the point on the diagram where the small perturbation to the input signal is introduced and an output point is the point on the diagram where the small perturbation to the output signal is measured.

The region between the input and output points defines the portion of the model that you want to linearize, unless a feedback loop feeds the output signal back into the input signal. When such a feedback loop is present, you might want to remove the effect of the loop by inserting an open loop point. Instructions for inserting open loop points are given in “Performing Open Loop Analysis” on page 3-13.

Note Linearization input and output points should not be confused with Simulink Inport and Outport blocks. Input and output points define linear analysis inputs and outputs, while Inport and Outport blocks define the operating point of the system.

Selecting Linearization Points

You can use Simulink Control Design to linearize the whole model, or any blocks or subsystems within the model. Define the system you are linearizing by placing linearization points before and after it in the model diagram.

In the magnetic ball model, the nonlinearities are all contained within the Magnetic Ball Plant. To linearize this subsystem,

1 Select the **Linearizations** node within the Control and Estimation Tools Manager.

2 On the model diagram position the mouse on the signal line between the Magnetic Ball Plant and the Controller. Right-click and select **Linearization Points -> Input Point** from the menu.

The model diagram now contains a small arrow pointing towards a circle just above the signal line between the Controller and the Magnetic Ball Plant, as in the following figure. This symbol indicates an input point for linearization has been placed there.

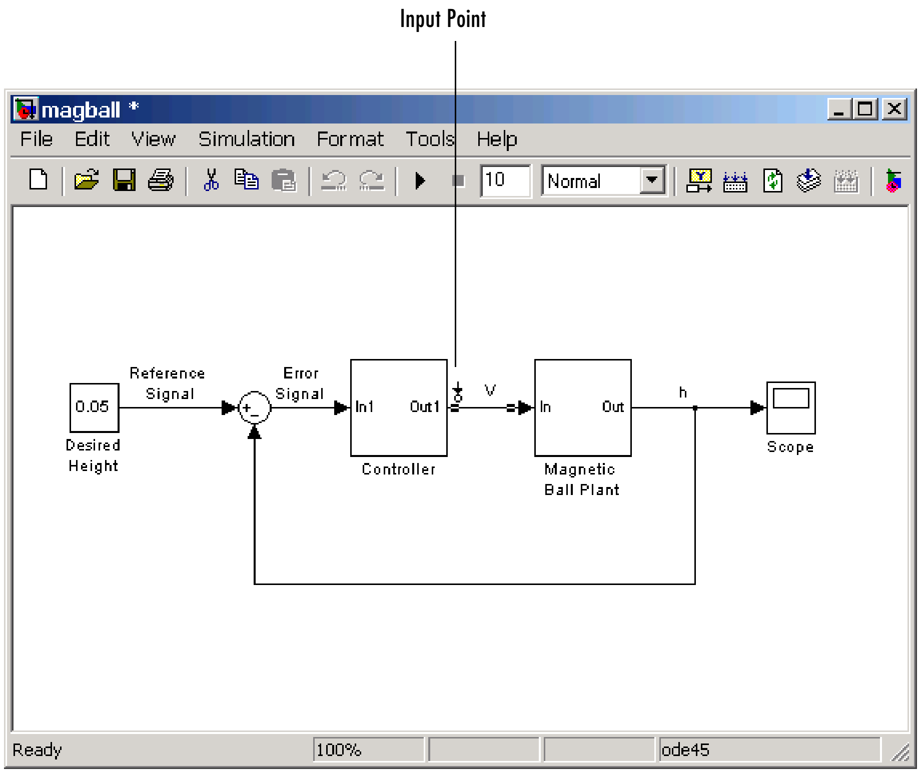


Figure 3-6: Magnetic Ball Model with Input Point Selected

3 Right-click the signal line after the Magnetic Ball Plant and select **Linearization Points -> Output Point** from the menu.

A small arrow pointing away from a circle on the signal line appears after the Magnetic Ball Plant indicating an output point for linearization has been placed there. The diagram should now look like that in the following figure.

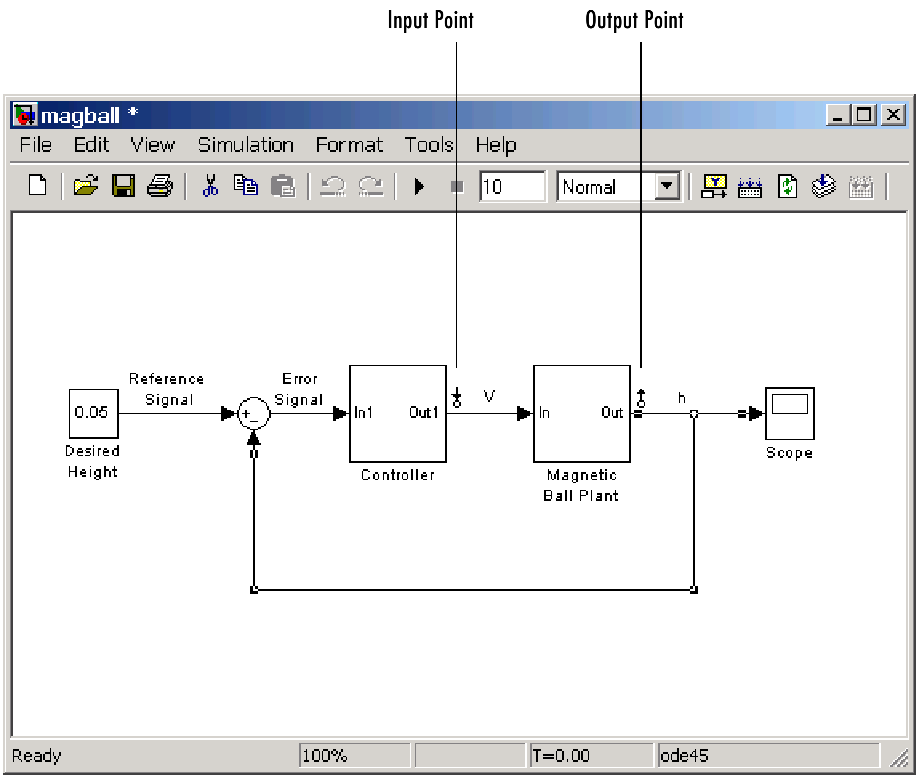


Figure 3-7: Magnetic Ball Model with Input and Output Points Selected

---

Note All linearization points are referenced to the output port of the block the signal line originates from. For example, placing a linearization point anywhere on the feedback loop in the figure above will result in a linearization point at the output of the Magnetic Ball Plant block.

Removing Linearization Points

To remove a linearization point from a signal line in your model, repeat the same process as for inserting a linearization point. For example, to remove an input point, right-click on the signal line containing the input point and select **Linearization Points -> Input Point** from the menu. The input point disappears from the diagram.

Performing Open Loop Analysis

Due to the presence of feedback loops in a model, the input and output points might not completely define the portion of the model you want to linearize. In these cases, to remove the effect of signals feeding back into the portion of the model you are linearizing, you might choose to insert an open loop point. For more information on this topic refer to “Understanding Open Loop Analysis” on page 2-11.

To linearize only the Magnetic Ball Plant, right-click the signal line containing the output point and choose **Linearization Points -> Open Loop**. This inserts a small $\times$ next to the output point in the diagram, representing a loop opening.

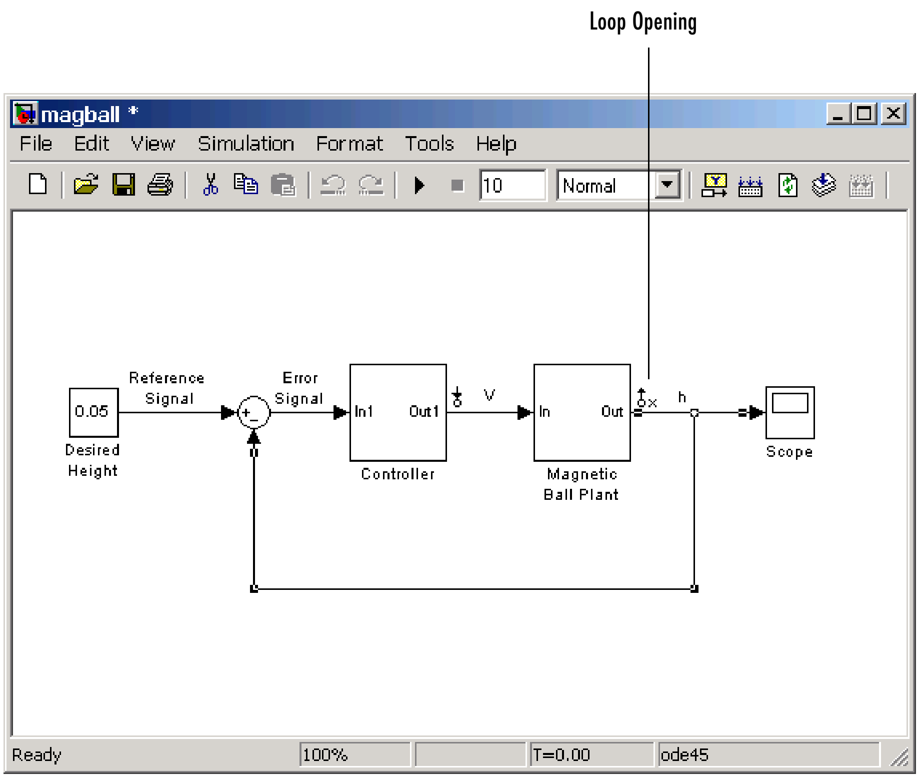


Figure 3-8: Magnetic Ball Model with Loop Opening

Note You do not need to place a loop opening in the same place as an input or output point. For example, in the following figure the highlighted blocks are included in the linearization. The loop opening is placed after the gain on the outer feedback loop, which removes the effect of this loop from the linearization. Placing a loop opening at the same place as the output point would have removed the effect of the inner loop from the linearization as well.

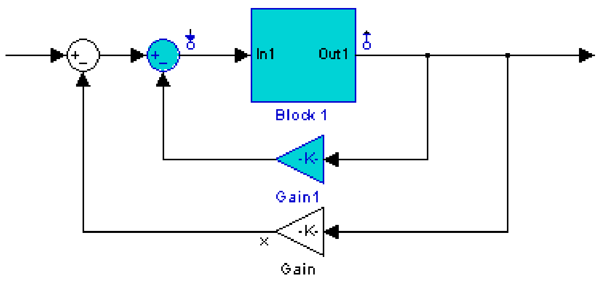
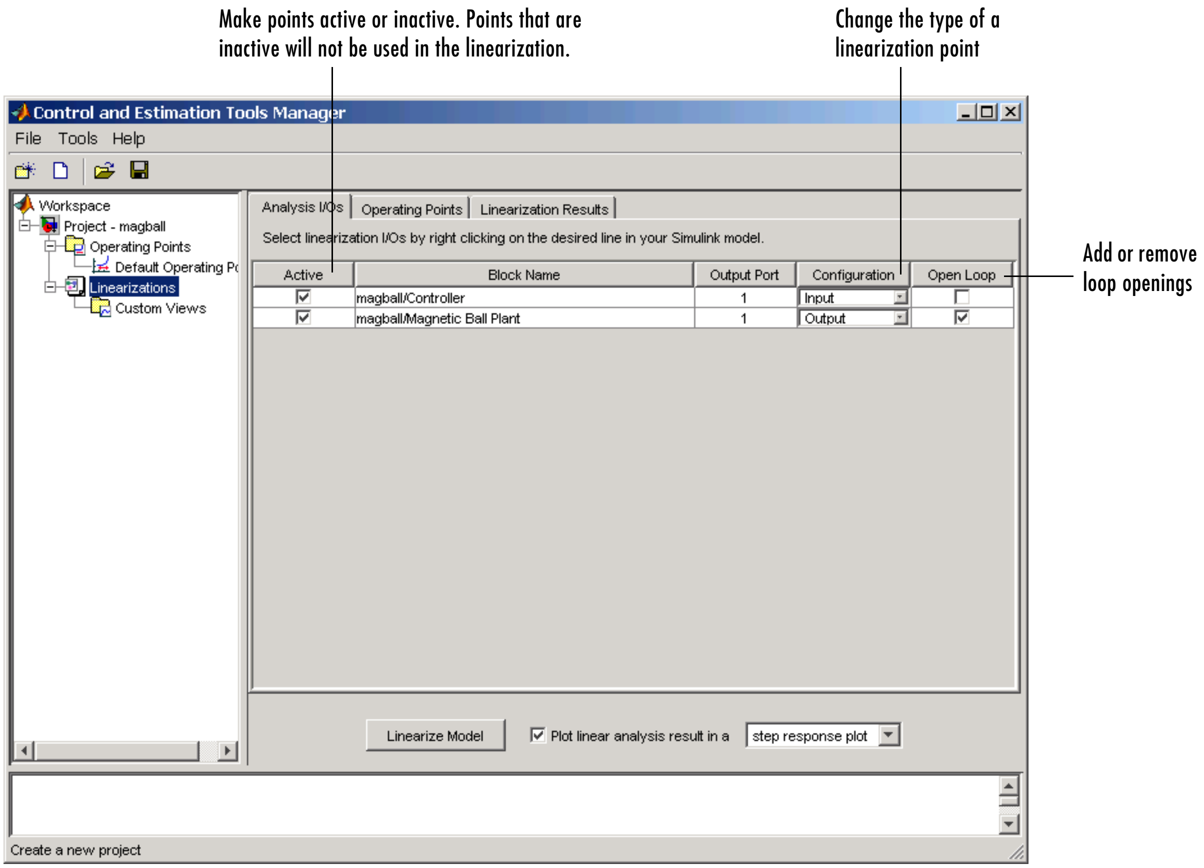


Figure 3-9: Open Loop Example

Inspecting Analysis I/Os

To view the linearization points, select the **Analysis I/Os** panel of the **Linearizations** node in the Control and Estimation Tools Manager, as shown in the following figure. Use this panel to inspect and make changes to your linearization points.


Make points active or inactive. Points that are inactive will not be used in the linearization.
Change the type of a linearization point
Add or remove loop openings
Control and Estimation Tools Manager
File Tools Help
Workspace
Project - magball
Operating Points
Default Operating Pc
Linearizations
Custom Views
Analysis I/Os
Operating Points
Linearization Results
Linearize Model
Create a new project

Select linearization I/Os by right clicking on the desired line in your Simulink model.

Active	Block Name	Output Port	Configuration	Open Loop
✓	magball/Controller	1	Input	☐
✓	magball/Magnetic Ball Plant	1	Output	✓

✓ Plot linear analysis result in a step response plot

Figure 3-10: Analysis I/Os Panel

Note You cannot make changes to the **Analysis I/Os** or perform open loop analysis when you select `Numerical Perturbation` as the **Linearization Algorithm** in the Linearization Task Options window (accessed by selecting **Tools -> Options** from the Control and Estimation Tools Manager window),. See "Linearizing Discrete-Time and Multi-Rate Models" on page 3-31 for information.

Specifying Operating Points

Before linearizing the model, you must choose an operating point about which to linearize the system. This is often a steady state value. Refer to "Operating Points" on page 2-3 for more information on the role of operating points in linearization.

There are five methods available for specifying the operating point:

- Completely specify all inputs and states in the operating point (page 3-19).

  For example, when you know that the height of the ball in `magball` should be 0.05, the rate of change of the height should be 0, the current should be 7.0036, and you also know the values of the states in the Controller, this information completely specifies the operating point.
- Specify target values or constraints on a subset of the model's inputs, outputs, and states (page 3-21). Simulink Control Design uses numerical methods to determine the full operating point based on this partial specification.

  For example, when you know that the height of the ball in `magball` should be 0.05, the rate of change of the height is small, the current should be positive, and your initial guess for the states in the Controller is 0, compute an operating point that closely matches the specifications.
- Extract an operating point from a simulation of the model (page 3-25).

  For example, you run a simulation of a model and use the values of the states and inputs at time 10 as the operating point values. This is especially useful when the simulation has reached a steady state.
- Accept the default operating point. The initial values of the states, inputs, and outputs, define this operating point. Only use the default operating point when the initial values are very close to the operating point of interest.
- Import a previously saved operating point from another project, the MATLAB workspace, or a file (page 3-27).

The following sections give instructions for each of these methods, with the exception of accepting the default values, which does not require any further action.

Note The operating point consists of values for *all* the states in the model although only those states between the linearization points will be linearized. This is because the whole model contributes to the operating point values of the states/inputs/outputs of the portion of the model you are linearizing.

Specifying Completely Known Operating Points

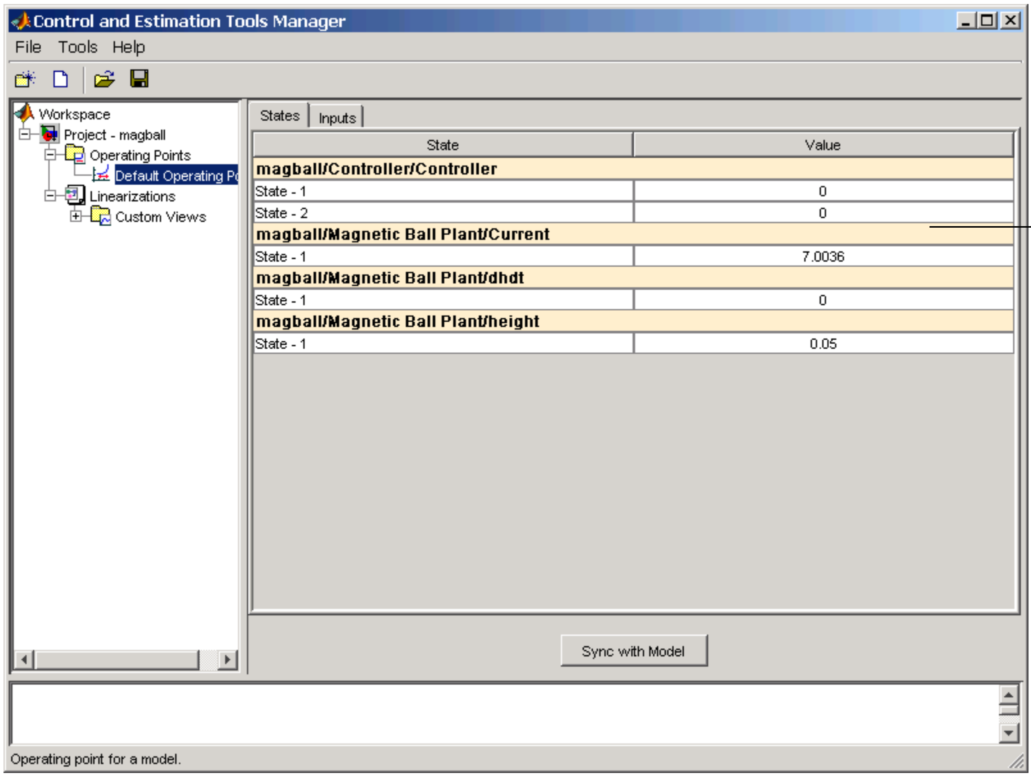
When you know the values of *all* states and inputs at the operating point, specify the operating point by editing the default operating point:

1 View the details of **Default Operating Point** by either

- Selecting it under the **Operating Points** node in the project tree,

  or

- Selecting it in the **Operating Points** panel of the **Linearizations** node, and then clicking **View**.



State	Value
magball/Controller/Controller	
State - 1	0
State - 2	0
magball/Magnetic Ball Plant/Current	
State - 1	7.0036
magball/Magnetic Ball Plant/dhdt	
State - 1	0
magball/Magnetic Ball Plant/height	
State - 1	0.05

Operating point for a model.

Figure 3-11: Default Operating Point

2 Edit the operating point by entering new values in the table. Switch between state and input values using the tabs on the left.

For example, change the value of **State-1** of **Controller** to `0.001` and the value of **Current** to `7`. Note that for this model, input values cannot be entered because the model determines all inputs.

When you add states, inputs, or outputs to the model, or remove them from the model, click the **Sync with Model** button to update the operating point table to reflect these changes.

Computing Operating Points from Specifications

Use Simulink Control Design to compute the operating point from specifications when you only know partial or implicit information:

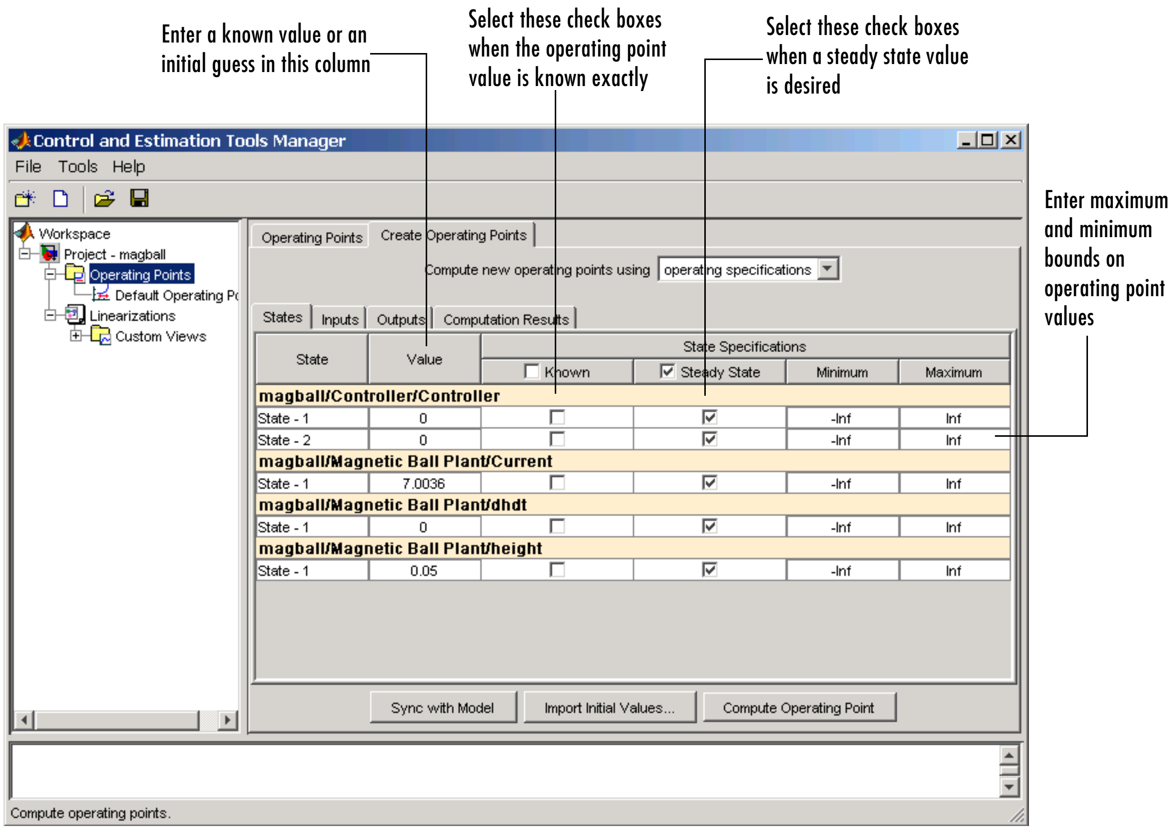
1 Create a new operating point by either

- Selecting the **Operating Points** node and then selecting the **Create Operating Points** tab on the right,

  or

- Clicking the **New Operating Point** button within the **Operating Points** panel of the **Linearizations** node.

2 From **Compute new operating points using**, select `operating specifications`. The Control and Estimation Tools Manager window should now look like that in the following figure.


Enter a known value or an initial guess in this column
Select these check boxes when the operating point value is known exactly
Select these check boxes when a steady state value is desired
Enter maximum and minimum bounds on operating point values
Control and Estimation Tools Manager
File Tools Help
Workspace
Project - magball
Operating Points
Default Operating Po
Linearizations
Custom Views
Operating Points
Create Operating Points
States
Inputs
Outputs
Computation Results
Sync with Model
Import Initial Values...
Compute Operating Point

Compute new operating points using: operating specifications

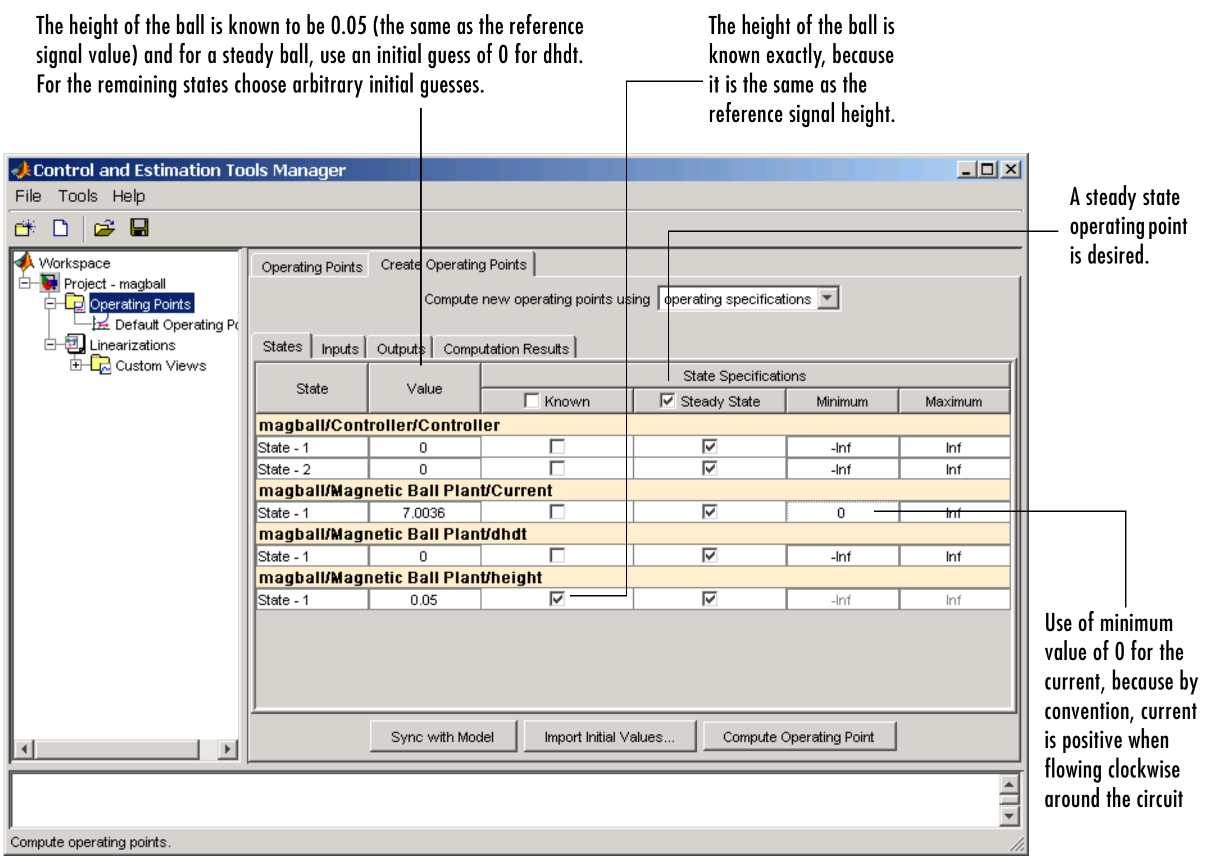
State	Value	State Specifications			
		☐ Known	✓ Steady State	Minimum	Maximum
magball/Controller/Controller					
State - 1	0	☐	✓	-Inf	Inf
State - 2	0	☐	✓	-Inf	Inf
magball/Magnetic Ball Plant/Current					
State - 1	7.0036	☐	✓	-Inf	Inf
magball/Magnetic Ball Plant/dhdt					
State - 1	0	☐	✓	-Inf	Inf
magball/Magnetic Ball Plant/height					
State - 1	0.05	☐	✓	-Inf	Inf

Compute operating points.

Figure 3-12: Create Operating Points Panel with Operating Specifications Selected

3 Enter operating point specifications in the table, such as any known values or constraints on signal values. Switch between states, inputs, and outputs using the tabs on the left.

A suitable set of specifications for the `magball` model is shown in the following figure. Note that in this case, the model determines the values of inputs and outputs.



The height of the ball is known to be 0.05 (the same as the reference signal value) and for a steady ball, use an initial guess of 0 for dhdt. For the remaining states choose arbitrary initial guesses.

The height of the ball is known exactly, because it is the same as the reference signal height.

A steady state operating point is desired.

Use of minimum value of 0 for the current, because by convention, current is positive when flowing clockwise around the circuit

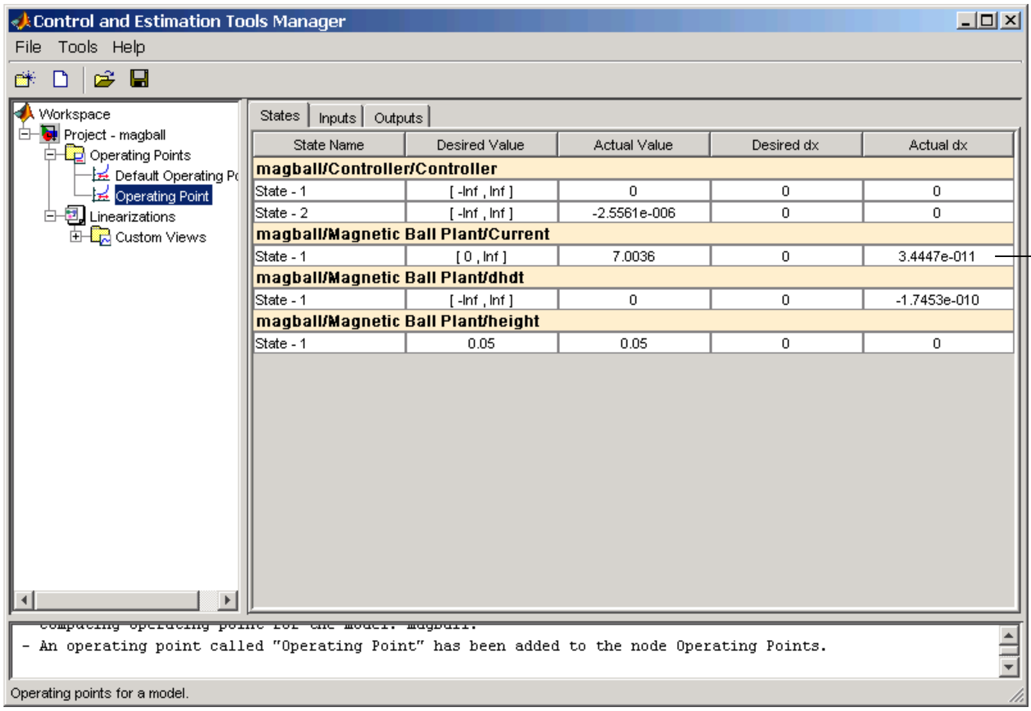
Compute new operating points using: operating specifications

State	Value	State Specifications			
		Known	✓ Steady State	Minimum	Maximum
magball/Controller/Controller					
State - 1	0	☐	✓	-Inf	Inf
State - 2	0	☐	✓	-Inf	Inf
magball/Magnetic Ball Plant/Current					
State - 1	7.0036	☐	✓	0	Inf
magball/Magnetic Ball Plant/dhdt					
State - 1	0	☐	✓	-Inf	Inf
magball/Magnetic Ball Plant/height					
State - 1	0.05	✓	✓	-Inf	Inf

Figure 3-13: Operating Point Specifications

When you add states, inputs, or outputs to the model, or remove them from the model, click the **Sync with Model** button to update the operating point table to reflect these changes.

4 Click **Compute Operating Point**. Simulink Control Design finds an operating point that closely matches the specifications and adds the new operating point, labeled **Operating Point**, to the **Operating Points** node. Note that some specifications, even values specified as **Known**, may not be met exactly. Select the operating point in the project tree to view its contents and assess the results.



State Name	Desired Value	Actual Value	Desired dx	Actual dx
magball/Controller/Controller				
State - 1	[-Inf , Inf]	0	0	0
State - 2	[-Inf , Inf]	-2.5561e-006	0	0
magball/Magnetic Ball Plant/Current				
State - 1	[0 , Inf]	7.0036	0	3.4447e-011
magball/Magnetic Ball Plant/dhdt				
State - 1	[-Inf , Inf]	0	0	-1.7453e-010
magball/Magnetic Ball Plant/height				
State - 1	0.05	0.05	0	0

```
 computing operating point for the model: magball.
- An operating point called "Operating Point" has been added to the node Operating Points.
```

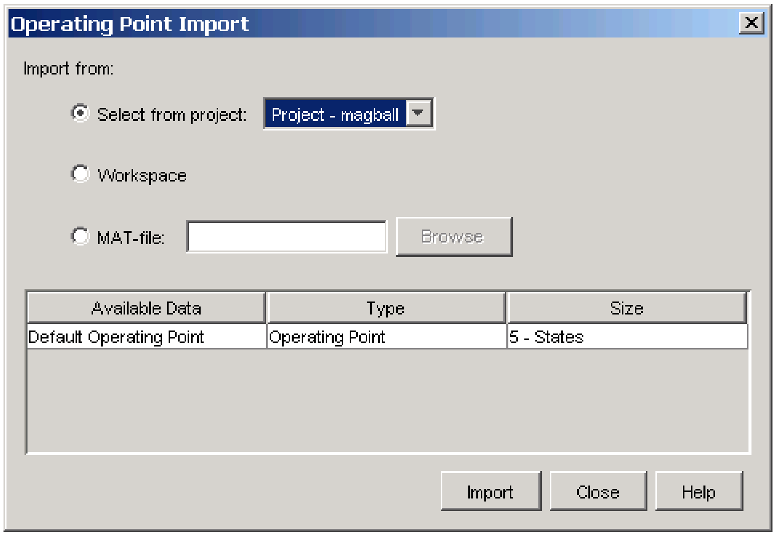
Operating points for a model.

Figure 3-14: Operating Point Results

For information on options that you can set when finding operating points from partial specifications, see “Extracting Operating Points from Simulation” on page 3-25 in the online documentation.

Importing Initial Values

When you want populate the **Value** column of the operating point specifications by importing initial or known values from another operating point, a Simulink states structure, or a vector of values, click the **Import Initial Values** button at the bottom of the window. The **Operating Point Import** dialog opens, as shown below.



[✓ Select from project:] Project - magball

Available Data	Type	Size
Default Operating Point	Operating Point	5 - States

Figure 3-15: Importing Operating Point Initial Values

Within this window select where to import the initial values from (a project, the workspace, or a file), then select the operating point from the list of available operating points below (or in the case of MAT-files, browse for a file). Click **Import** to import the initial values from the selected operating point into the **Value** column of the operating point specifications.

Constraining Outputs

Operating specifications often include constraints on the values of specific signals in the model. To constrain output signals when determining operating points from specifications, add an output constraint annotation to the model by right-clicking on the signal line and choosing **Output Constraint** from the menu. This adds a small T to the signal line. The **Outputs** panel within the **Create Operating Points** panel then allows you to enter desired values as well as minimum and maximum values for this signal.

Extracting Operating Points from Simulation

Use Simulink Control Design to extract operating points from a simulation of your model at specified times, such as when the simulation reaches a steady state solution:

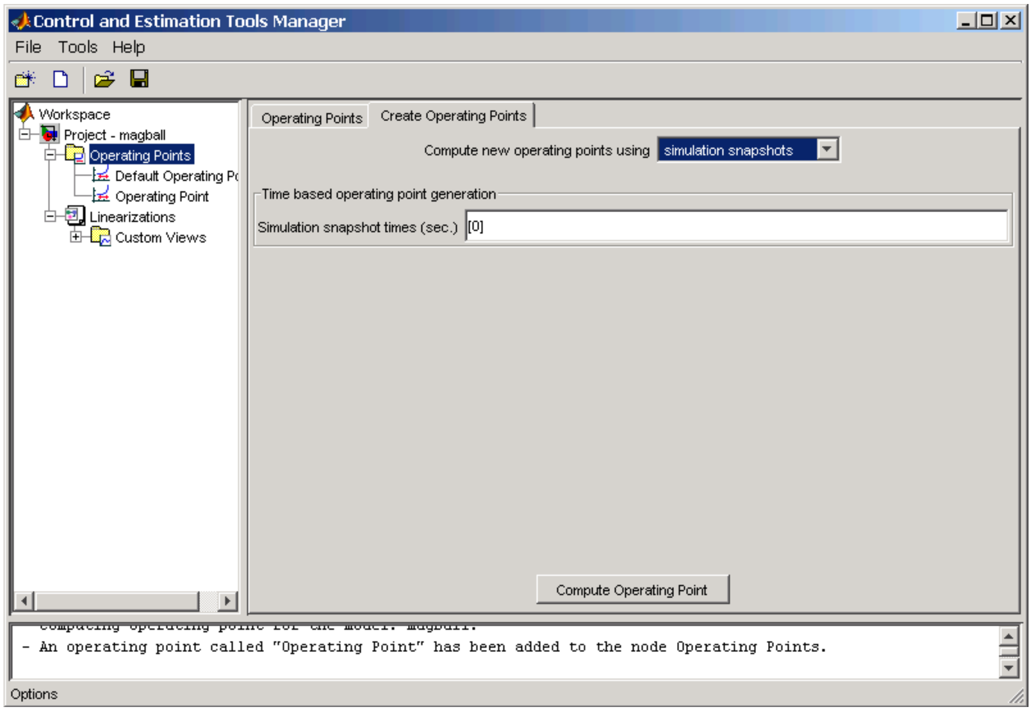
1 Create a new operating point by either

- Selecting the **Operating Points** node in the project tree and then selecting the **Create Operating Points** tab on the right,

  or

- Clicking the **New Operating Point** button within the **Operating Points** panel of the **Linearizations** node.

2 From **Compute new operating points using**, select `simulation snapshots`. The window should now look like that in this figure.



Compute new operating points using: simulation snapshots

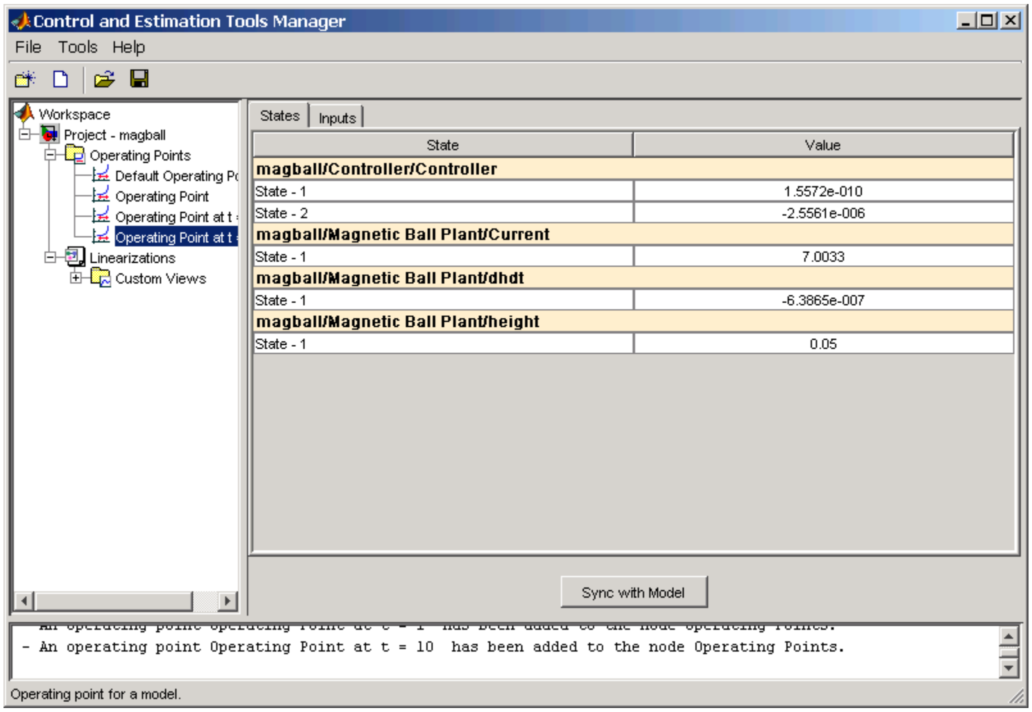
Simulation snapshot times (sec.): [0]

```
- An operating point called "Operating Point" has been added to the node Operating Points.
```

Figure 3-16: Simulation Snapshots Set-Up

3 Enter a vector of times in the **Simulation Snapshot Times** field. For example, to compute operating points at `t=1` and `t=10`, enter `[1,10]`.

4 Click **Compute Operating Point**. Simulink Control Design simulates the model, extracts operating points, labeled **Operating Point at t=1** and **Operating Point at t=10**, and adds them to the **Operating Points** node in the project tree. Select an operating point to view its contents and assess the results as shown in this figure.



State	Value
magball/Controller/Controller	
State - 1	1.5572e-010
State - 2	-2.5561e-006
magball/Magnetic Ball Plant/Current	
State - 1	7.0033
magball/Magnetic Ball Plant/dhdt	
State - 1	-6.3865e-007
magball/Magnetic Ball Plant/height	
State - 1	0.05

```
- An operating point Operating Point at t = 10  has been added to the node Operating Points.
```

Operating point for a model.

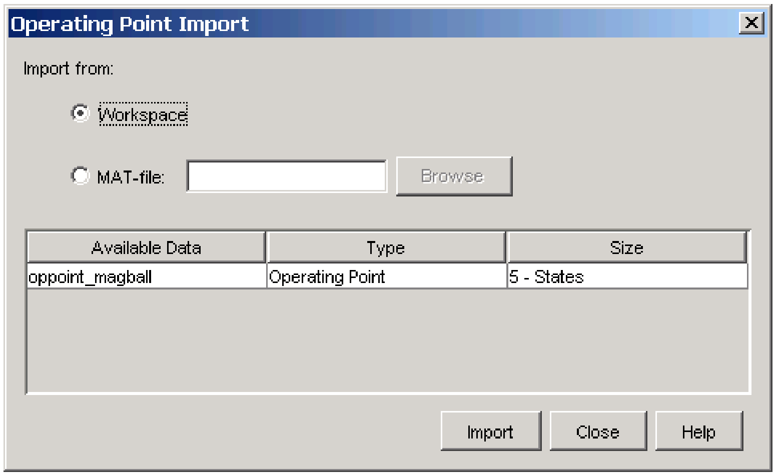
Figure 3-17: Simulation Snapshot Operating Point Results

When you add states, inputs, or outputs to the model, or remove them from the model, click the **Sync with Model** button to update the operating point table to reflect these changes.

Importing Operating Points

Use Simulink Control Design to import operating points from the MATLAB workspace or from a MAT-file.

1 To import a new operating point, select the **Operating Points** node in the project tree and then select the **Operating Points** tab on the right. Click the **Import** button at the bottom of the panel. This displays the **Operating Point Import** dialog as shown in the figure below.



Available Data	Type	Size
oppoint_magball	Operating Point	5 - States

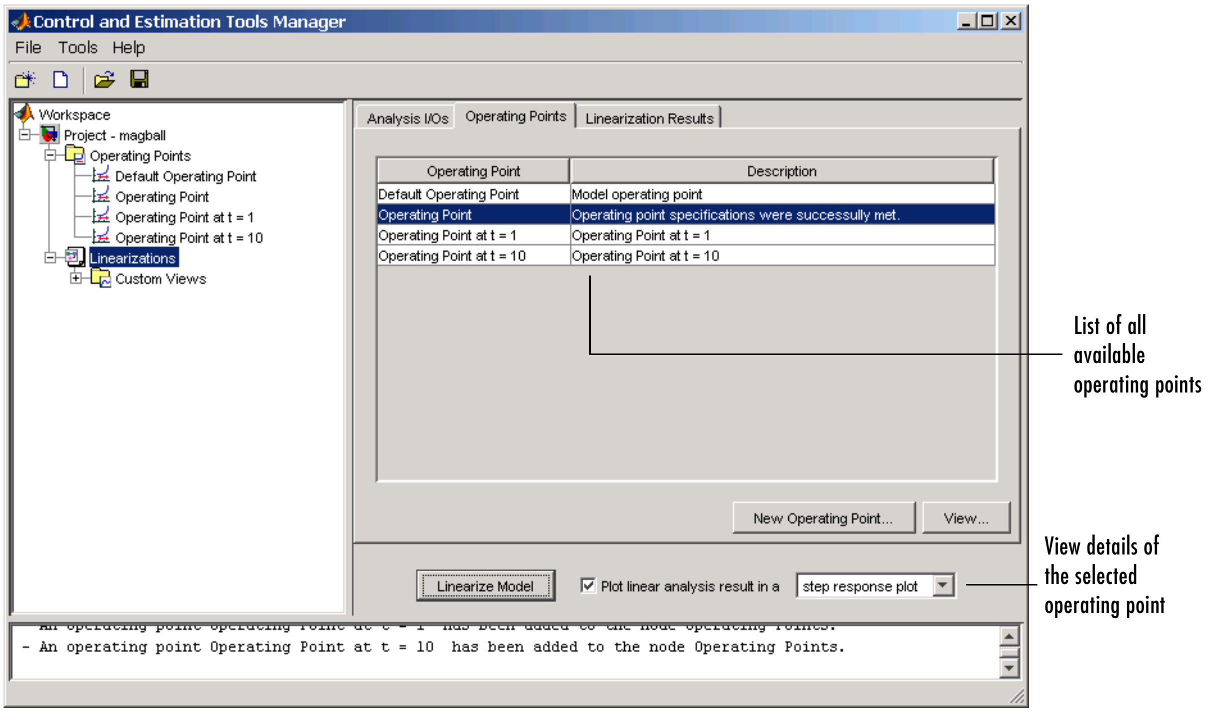
Figure 3-18: Operating Point Import Dialog

2 Within the **Operating Point Import** dialog choose **Workspace** or **MAT-file** as the location to import the operating point from, select an operating point from the list below, and then click **Import**.

Linearizing the Model

After configuring the model and setting up the operating points, you are ready to linearize the model.

Within the **Linearizations** node, select the **Operating Points** tab. A list of all available operating points for this project is shown. Select the operating point(s) you want to use for the linearization. To linearize the magnetic ball model around the operating point computed from partial specifications, select that line in the list, as shown in this figure.



Operating Point	Description
Default Operating Point	Model operating point
Operating Point	Operating point specifications were successully met.
Operating Point at t = 1	Operating Point at t = 1
Operating Point at t = 10	Operating Point at t = 10

☑ Plot linear analysis result in a: step response plot

```
- An operating point Operating Point at t = 10  has been added to the node Operating Points.
```

Figure 3-19: Select Operating Point for Linearization

To compute the linearized model, click the **Linearize Model** button at the bottom of the panel. The results display, by default, as a step response of the new linearized system in the LTI Viewer window, as shown in the following figure.

Select from one of seven options in the menu to the right of the **Linearize Model** button to display the results in a different format. Clear the check box next to this menu and the results will not display in the LTI Viewer.

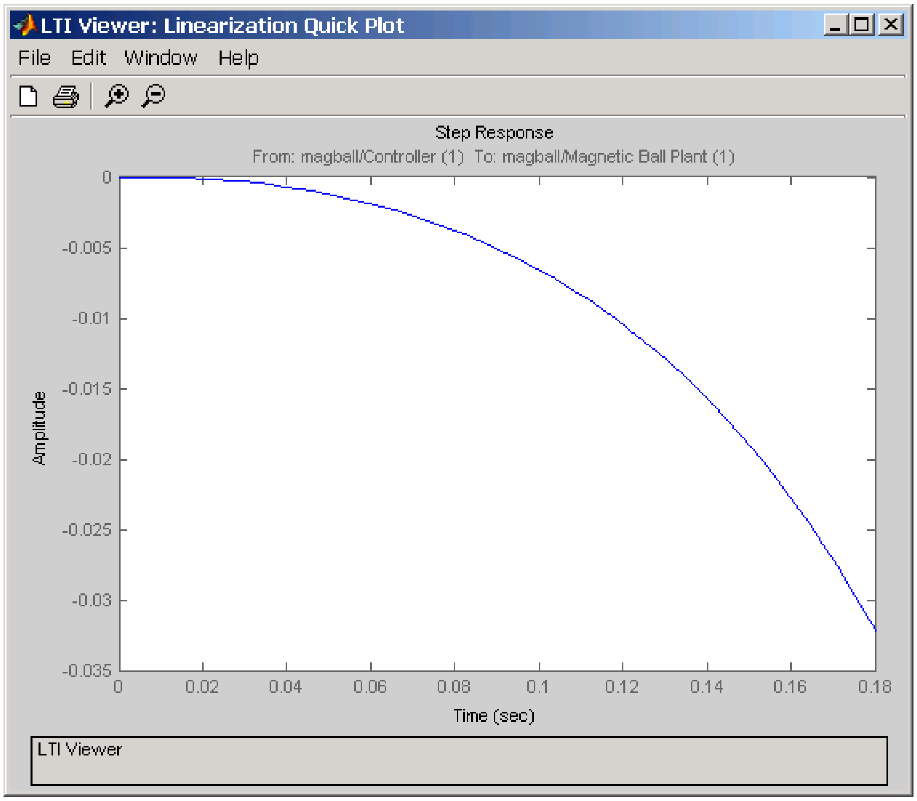


Figure 3-20: Linearization Results in LTI Viewer

For information on setting options for linearization, refer to the online documentation.

Using the LTI Viewer

To add characteristics such as settling time or peak response to your plot, right-click anywhere in the plot area and choose an option from the menu. Add data markers by clicking the point you want to mark. In addition, you can display up to six plots at one time. To change the number of plots select **Edit -> Plot Configurations**. Export the linearized model to the workspace by selecting **File -> Export**. For more information on the LTI Viewer, please refer to the Control System Toolbox documentation.

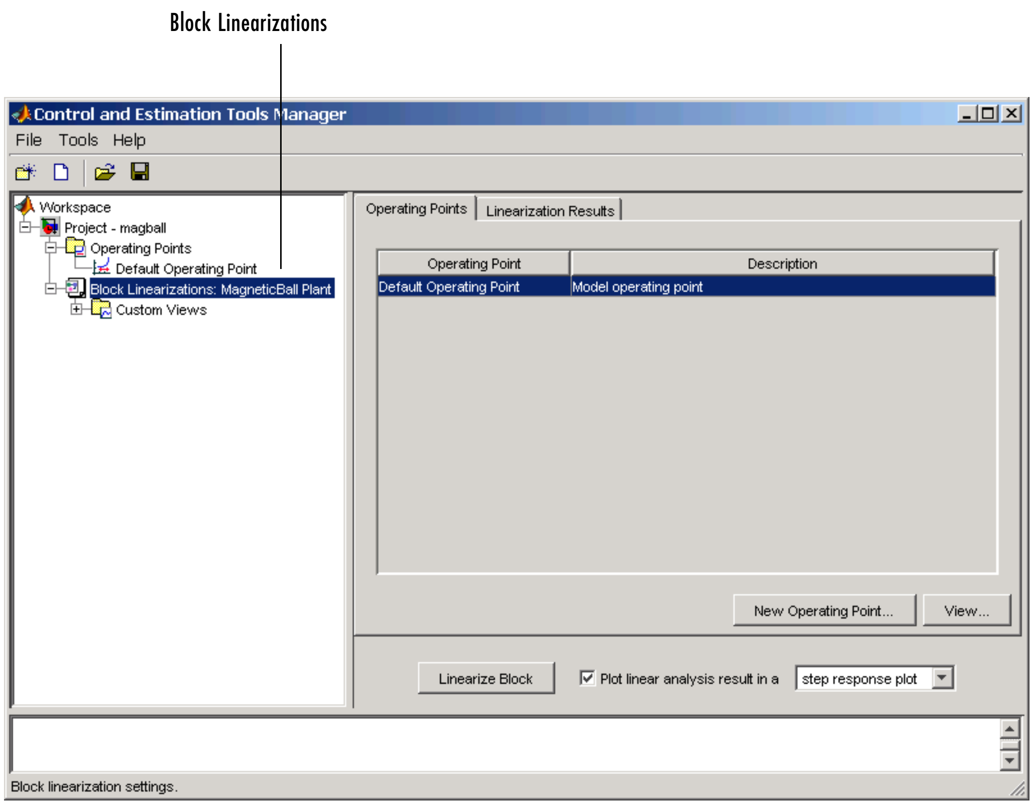
Linearizing Discrete-Time and Multi-Rate Models

The linearization method is the same for models containing discrete-time states or several different sample times. However, you can choose to adjust the **Linearization sample time** in the **Linearization** options panel. By default, this parameter is set to `-1`, in which case Simulink Control Design linearizes at the slowest sample rate in the model. To create a linearized model with a different sample time, enter a new value in the dialog window. A value of `0` gives a continuous-time model.

Linearizing a Block

With Simulink Control Design you can also linearize a single block in a Simulink model. To do this, right-click the block and choose **Linearize Block** from the context menu.

This adds a **Block Linearizations** node to the project tree as shown in the following figure.



Operating Point	Description
Default Operating Point	Model operating point

☑ Plot linear analysis result in a step response plot

Block linearization settings.

Figure 3-21: Control and Estimation Tools Manager with Block Linearizations

To complete the linearization, specify an operating point in the same way as when linearizing models, and then click the **Linearize Block** button within the

Operating Points panel of the **Block Linearizations** node. You do not need to choose linearization input and output points because the inports and outports of the block are used. If you do have linearization input and output points in your model, they will be ignored. You cannot linearize an individual block using numerical-perturbation linearization (when `Numerical perturbation` is selected as the **Linearization Algorithm** parameter).

Note To linearize an individual block, it must contain at least one data inport and outport. Since SimMechanics and SimPowerSystems blocks have connection ports instead of inports and outports, they cannot be individually linearized.

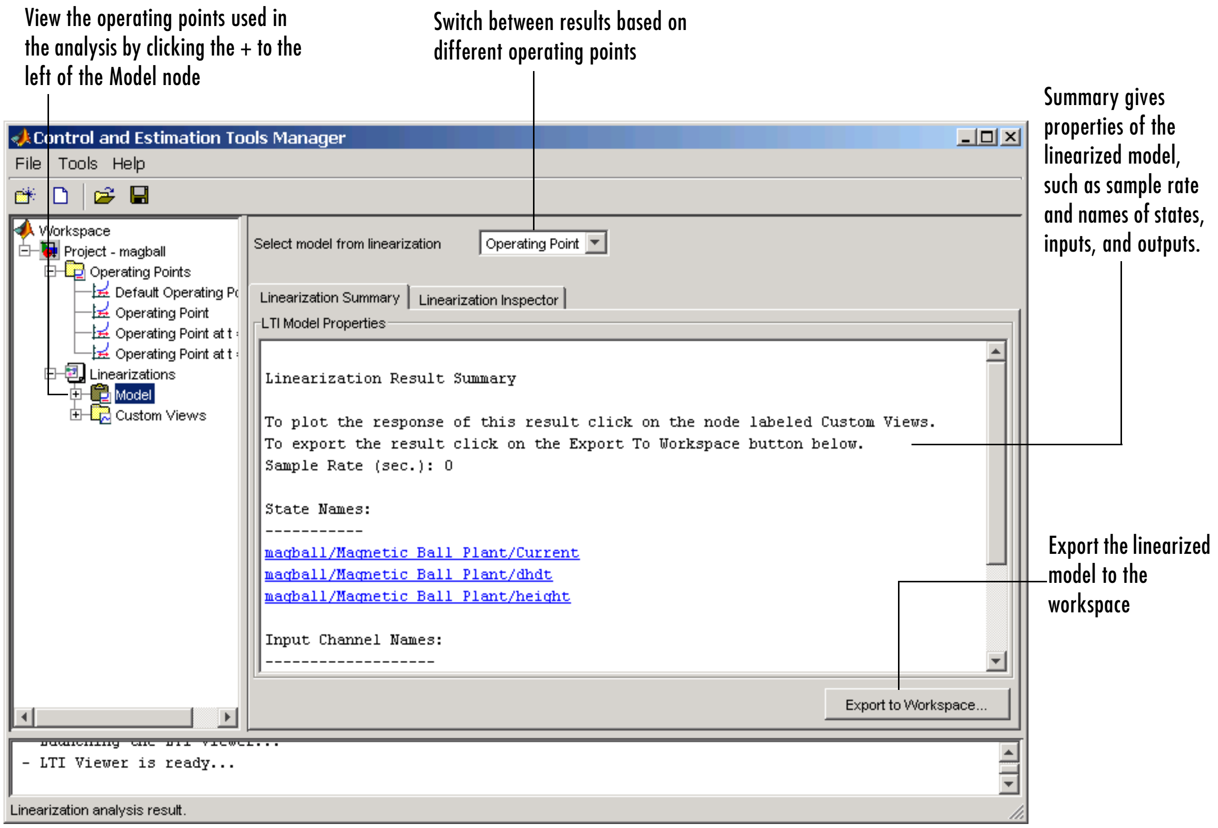
Analyzing the Results

To display all the linearizations of the current project, select the **Linearization Results** tab within the **Linearizations** node. At this stage you should have just one, **Model**, which also appears as its own node in the project tree. To remove a linearization from the **Linearization Results** list, select it and then use the **Delete** button at the bottom of the panel.

To inspect and validate the results, refer to "Validating the Linearization" on page 3-34. To display and configure plots of the results, refer to "Displaying the Results" on page 3-38. For methods on simulating the linearized model for comparison with the original model, refer to "Comparing the Linearized and Original Models" in the online documentation.

Validating the Linearization

To display the **Linearization Summary**, as shown in the following figure, click the **Model** node in the project tree. To delete the linearization result, right-click **Model** and select **Delete** from the menu.


View the operating points used in the analysis by clicking the + to the left of the Model node
Switch between results based on different operating points
Summary gives properties of the linearized model, such as sample rate and names of states, inputs, and outputs.
Export the linearized model to the workspace
Control and Estimation Tools Manager
File
Tools
Help
Workspace
Project - magball
Operating Points
Default Operating Po
Operating Point
Operating Point at t
Operating Point at t
Linearizations
Model
Custom Views
Select model from linearization
Operating Point
Linearization Summary
Linearization Inspector
LTI Model Properties
Export to Workspace...
Linearization analysis result.

```
Linearization Result Summary

To plot the response of this result click on the node labeled Custom Views.
To export the result click on the Export To Workspace button below.
Sample Rate (sec.): 0

State Names:
-----------
magball/Magnetic Ball Plant/Current
magball/Magnetic Ball Plant/dhdt
magball/Magnetic Ball Plant/height

Input Channel Names:
-------------------
```

```
- LTI Viewer is ready...
```

Figure 3-22: Linearization Summary Panel

Highlighting Blocks in the Linearization

To verify that the correct portion of the model was linearized, you can highlight the blocks in the model that were used in the linearization. To do this, right-click the **Model** node and select **Highlight Blocks in Linearization**. The following figure shows that the Magnetic Ball Plant block, including all blocks within it, was used in the linearization.

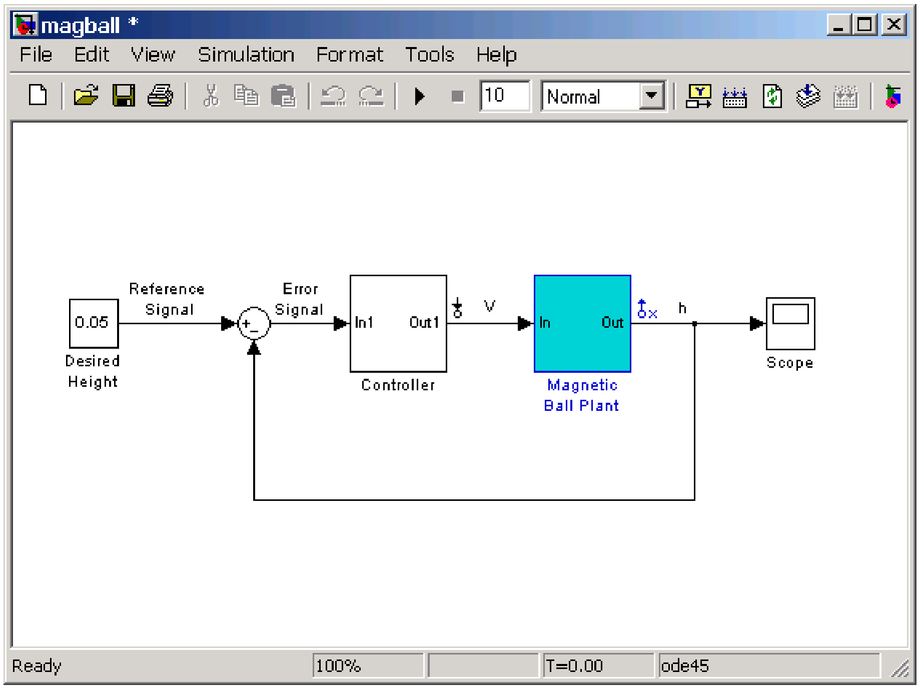


Figure 3-23: Highlight Blocks in Linearization Turned On

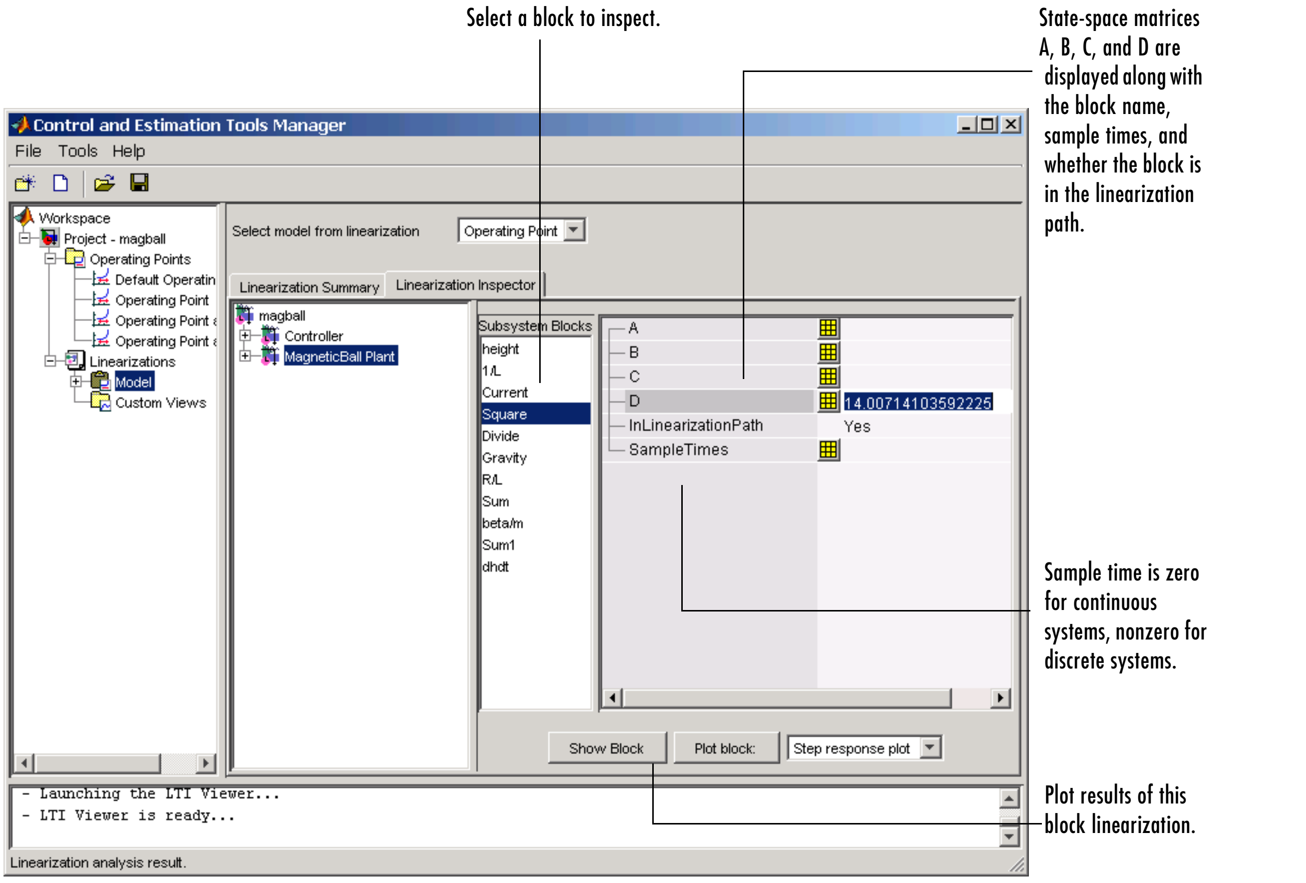
To remove the highlighting, right-click the **Model** node and select **Remove Highlighting**.

Note You cannot use the **Highlight Blocks in Linearization** or **Remove Highlighting** options for models that are linearized with numerical-perturbation linearization (when `Numerical perturbation` is selected as the **Linearization Algorithm** parameter in the Linearization Task Options window).

Inspecting the Linearization Block-by-Block

To inspect the linearization block-by-block, click the **Linearization Inspector** tab within the **Model** node. This pane displays the linearizations of the individual blocks in the linearization path, that is, the blocks between the input and output points. Use this information to determine if the linearization gave the expected results, and to compare linearizations using alternative operating points or parameter values.

This figure shows the results of the magnetic ball model in the **Linearization Inspector** pane with **Magnetic Ball Plant** highlighted.



Select a block to inspect.

State-space matrices A, B, C, and D are displayed along with the block name, sample times, and whether the block is in the linearization path.

Sample time is zero for continuous systems, nonzero for discrete systems.

Plot results of this block linearization.

Select model from linearization: Operating Point

D: 14.00714103592225

InLinearizationPath: Yes

```
- Launching the LTI Viewer...
- LTI Viewer is ready...
```

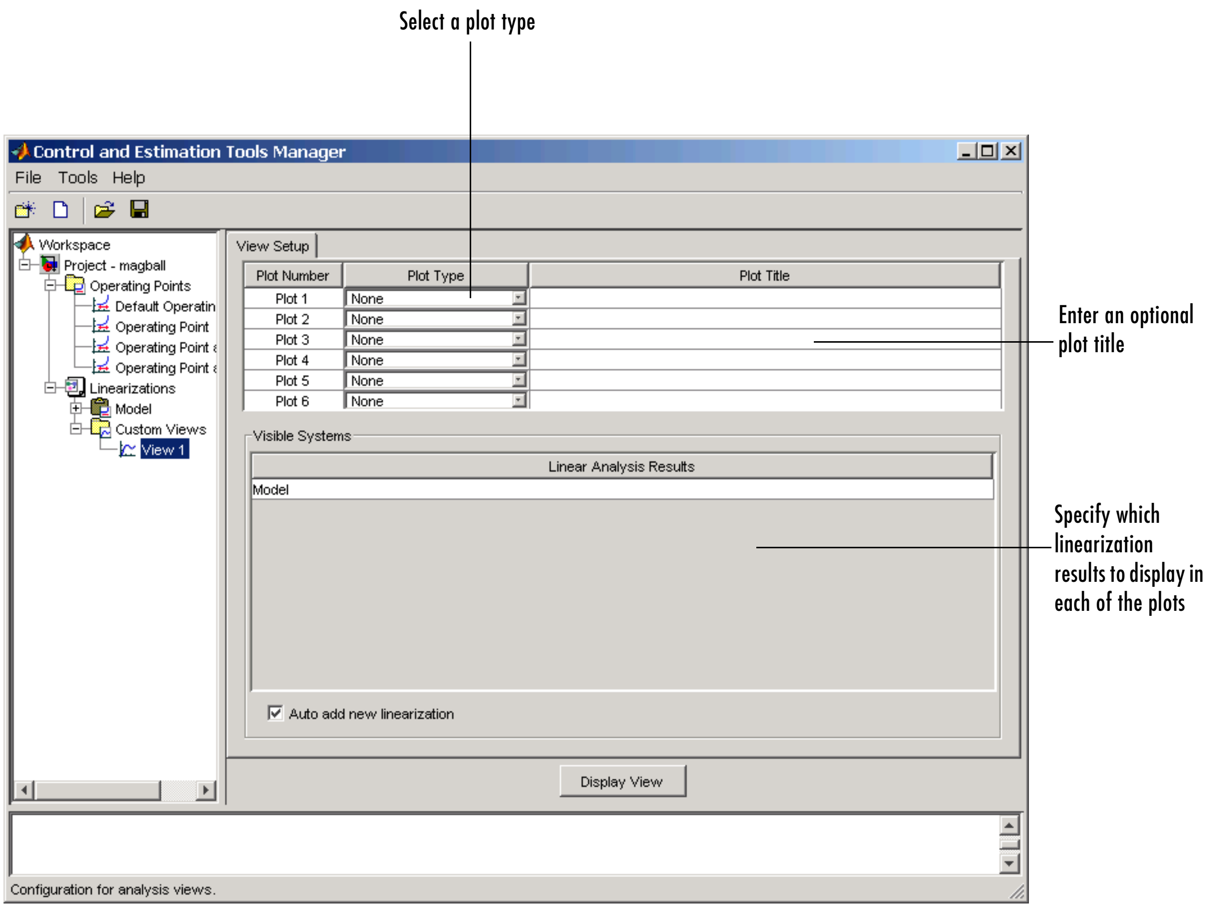
Linearization analysis result.

Figure 3-24: Linearization Inspector Panel of Simulink Control Design

Note The **Linearization Inspector** pane does not exist for models that are linearized with numerical-perturbation linearization (when `Numerical perturbation` is selected as the **Linearization Algorithm** parameter in the **Linearization Task Options** window) as there are no individual block linearizations to inspect.

Displaying the Results

The LTI Viewer automatically displays a **Linearization Quick Plot** of the linearization results after the linearization. To create additional, customized plots, right-click the **Custom Views** node under the **Linearizations** node and select **Add View**. A new view, **View1** is added to the **Custom Views** node. Select **View1** to display the **View Setup** pane and set up this view.



Plot Number	Plot Type	Plot Title
Plot 1	None	
Plot 2	None	
Plot 3	None	
Plot 4	None	
Plot 5	None	
Plot 6	None	

Linear Analysis Results
Model

☑ Auto add new linearization

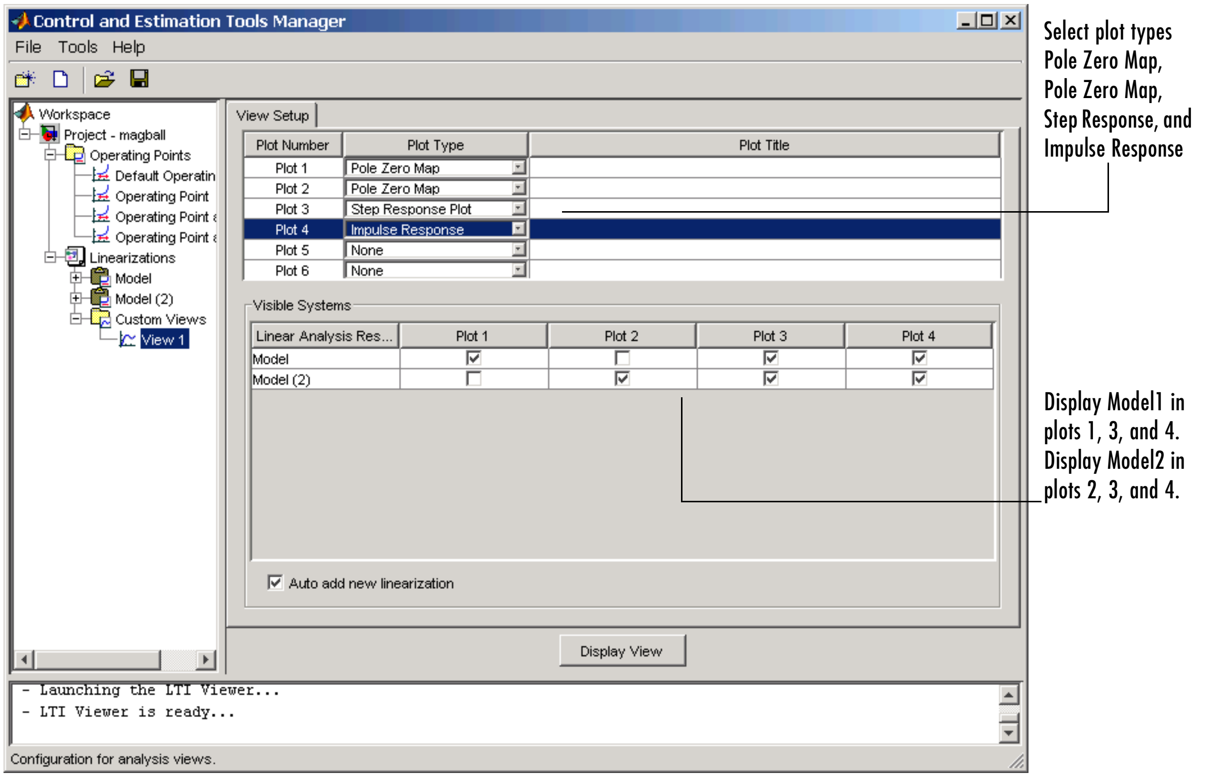
Configuration for analysis views.

Figure 3-25: View Setup Panel

Example

Before configuring the **View Setup** pane for the magnetic ball system, run another linearization of this model using the operating point at t=10 that you computed in "Extracting Operating Points from Simulation" on page 3-25. This adds **Model (2)** to the **Linearizations** node.

Configure **View Setup** as shown in the following figure.


Control and Estimation Tools Manager
File Tools Help
Workspace
Project - magball
Operating Points
Default Operatin
Operating Point
Operating Point
Operating Point
Linearizations
Model
Model (2)
Custom Views
View 1
View Setup
Visible Systems
Display View
Configuration for analysis views.
Select plot types Pole Zero Map, Pole Zero Map, Step Response, and Impulse Response
Display Model1 in plots 1, 3, and 4. Display Model2 in plots 2, 3, and 4.

Plot Number	Plot Type	Plot Title
Plot 1	Pole Zero Map	
Plot 2	Pole Zero Map	
Plot 3	Step Response Plot	
Plot 4	Impulse Response	
Plot 5	None	
Plot 6	None	

Linear Analysis Res...	Plot 1	Plot 2	Plot 3	Plot 4
Model	✓		✓	✓
Model (2)		✓	✓	✓

✓ Auto add new linearization

```
- Launching the LTI Viewer...
- LTI Viewer is ready...
```

Figure 3-26: Settings for Custom Plot Example

Click **Display View** to display the LTI Viewer, as shown in the following figure.

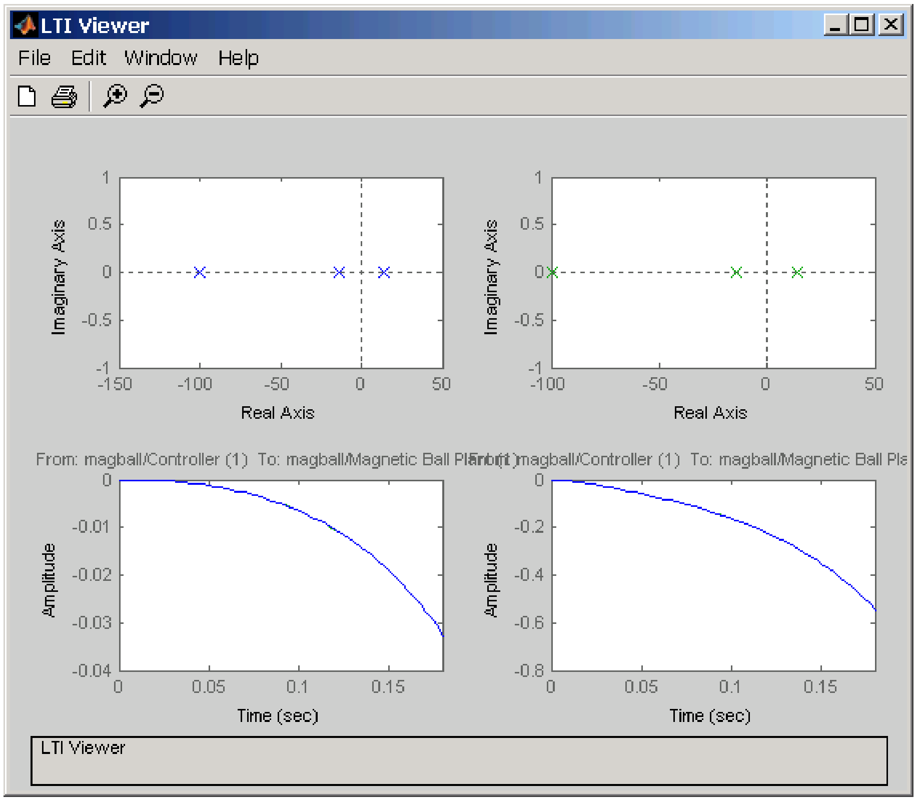

LTI Viewer
File Edit Window Help
Imaginary Axis
Real Axis
From: magball/Controller (1) To: magball/Magnetic Ball Plant (1)
From: magball/Controller (1) To: magball/Magnetic Ball Pla
Amplitude
Time (sec)
LTI Viewer

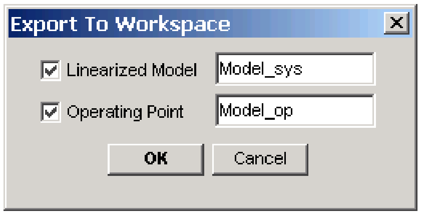
Figure 3-27: Customized View of Results

Exporting and Saving Your Work

The Control and Estimation Tools Manager provides the capability to export your linearization results to the MATLAB workspace, and to save and reopen your linearization projects.

Exporting Results

Export the operating point and results of the linearization to the MATLAB workspace for further analysis using command line functions. To export these results, right-click the results node, **Model,** under the **Linearizations** node and choose **Export** from the menu. In the **Export To Workspace** dialog box, choose new names for the linearized model and operating point, or accept the defaults, and then click **OK**.


Export To Workspace

☑ Linearized Model: Model_sys

☑ Operating Point: Model_op

Figure 3-28: Export Dialog Box

The MATLAB workspace now contains two new objects, `Model1_op` and `Model1_sys`. To see this, type

```
who
```

at the MATLAB prompt. This returns

```
Your variables are:

L           Model_sys   beta        m
Model_op    R           g
```

Alternatively, you can export the results to the MATLAB workspace by selecting **File -> Export** from the LTI Viewer window, or by clicking the **Export to Workspace** button at the bottom of the **Linearization Summary** panel within the **Model** node.

By right-clicking the results node, **Model**, you can also delete results.

Exporting and Restoring Linearization I/O Settings

To export linearization I/O settings to the MATLAB workspace, use the `getlinio` function. You can save these settings using the `save` function. Use them in a later session by reloading them with the `load` function. Upload them to the model diagram with the `setlinio` function.

For more information, see the function reference pages for `getlinio` and `setlinio`.

Saving Control and Estimation Tools Manager Projects

A Control and Estimation Tools Manager project can consist of multiple tasks including those from Simulink Control Design, Simulink Parameter Estimation, and the Model Predictive Control Toolbox. Each task contains data, objects, and results for the analysis of a particular model.

To save your projects, choose **File -> Save** in the Control and Estimation Tools Manager window.

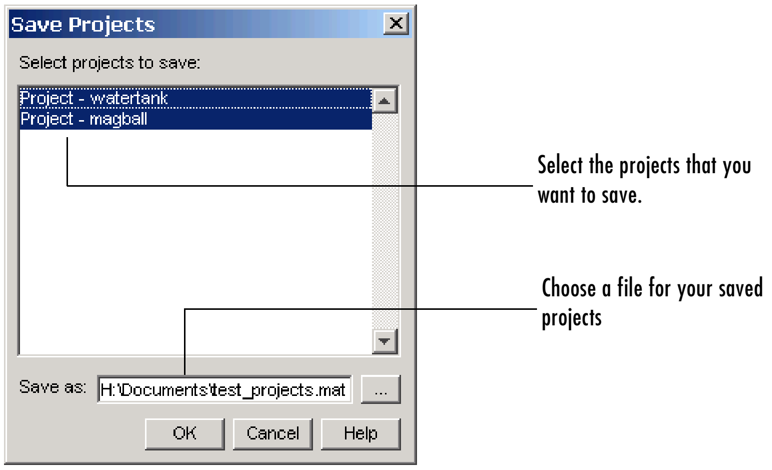

Save Projects
Select projects to save:
Project - watertank
Project - magball
Save as: H:\Documents\test_projects.mat
...
OK
Cancel
Help
Select the projects that you want to save.
Choose a file for your saved projects

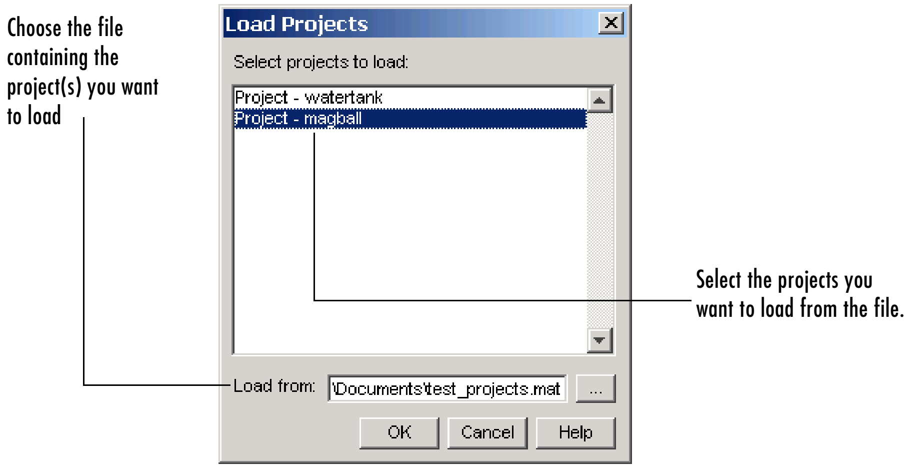
Figure 3-29: Save Projects Dialog

In the **Save Projects** dialog box, select the projects that you want to save. You can save multiple projects within one file. Next, choose a directory and name for your project file by either browsing for a file or typing the full path and filename in the **Save as** box, and then click **OK**. The project-file is saved as a MAT-file.

Opening Control and Estimation Tools Manager Projects

A Control and Estimation Tools Manager project can consist of multiple tasks including those from Simulink Control Design, Simulink Parameter Estimation, and the Model Predictive Control Toolbox. Each task contains data, objects, and results for the analysis of a particular model.

To open previously saved projects, choose **File -> Load** in the Control and Estimation Tools Manager window.


Choose the file containing the project(s) you want to load
Load Projects
Select projects to load:
Project - watertank
Project - magball
Load from:
\Documents\test_projects.mat
...
OK
Cancel
Help
Select the projects you want to load from the file.

In the **Load Projects** dialog box, choose a project-file by either browsing for the directory and file, or typing the full path and filename in the **Load from** box. Project-files are always MAT-files. The projects within this file appear in the list-box. Select the projects that you want to load, then click **OK**. When a file contains multiple projects you can choose to load them all or just a few.

Linearizing Models Using Functions

Use the Simulink Control Design linear analysis functions to linearize Simulink models in the MATLAB Command Window. Use the functions when you want to create M-files to automate the linearization process, or perform *batch* linearization.

Overview

As discussed in “Purpose of Linearization” on page 2-2, linearization is an important process in the design and analysis of control systems. The main steps involved in the linearization of Simulink models using the Simulink Control Design functions are

1. Creating or opening a Simulink model
2. Configuring the model
3. Specifying operating points
4. Linearizing the model
5. Analyzing the results and saving your work

The following section introduces an example containing a nonlinear system, a water-filled tank. Subsequent sections of this chapter use the example for a detailed discussion of each step.

Although this chapter focuses on the Simulink Control Design functions for linearizing models, you can also use the Graphical User Interface (GUI) for some steps in the process. For example, after specifying the operating points in the GUI, you can export the results to the MATLAB workspace and use the functions to continue the analysis. Chapter 3, “Linearizing Models,” gives detailed information on use of the GUI. For discussion of the advantages and disadvantages of the GUI versus the functions, refer to “Using the GUI vs. Command Line Functions” on page 1-2. A particular advantage of the linearization functions is the ability to write scripts to automate the linearization process or perform *batch linearization*.

Example: Water-Tank System

Water enters a tank from the top and leaves through an orifice in its base. The rate that water enters is proportional to the voltage, V, applied to the pump. The rate that water leaves is proportional to the square root of the height of water in the tank.

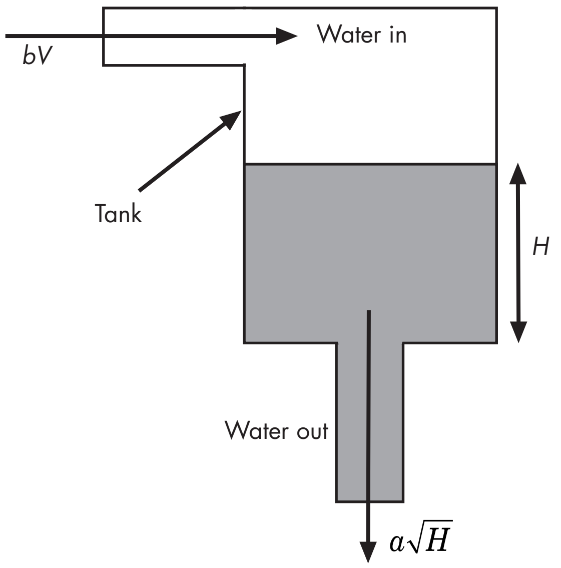


Figure 4-1: Schematic Diagram for the Water-Tank System

Model Equations

A differential equation for the height of water in the tank, H, is given by

$$\frac{d}{dt}Vol = A\frac{dH}{dt} = bV - a\sqrt{H}$$

where Vol is the volume of water in the tank, A is the cross-sectional area of the tank, b is a constant related to the flow rate into the tank, and a is a constant related to the flow rate out of the tank. The equation describes the height of water, *H*, as a function of time, due to the difference between flow rates into and out of the tank.

The equation contains one state, *H*, one input, *V*, and one output, *H*. It is nonlinear due to its dependence on the square-root of *H*. Linearizing the model, using Simulink Control Design, simplifies the analysis of this model. For information on the linearization process, see “Linearization of Nonlinear Models” on page 2-5.

Creating or Opening a Simulink Model

To begin linearization using functions, you must first create or open a Simulink model of your system. The model can have any number of inputs and outputs (including none) and any number of states. The model can include user-defined blocks or S-functions. Your model can involve a plant only, a plant with a feedback loop and controller, or any number of subsystems.

To continue with the water-tank example, type

```
watertank
```

at the MATLAB prompt. This opens a Simulink model containing the water-tank system as shown in this figure.

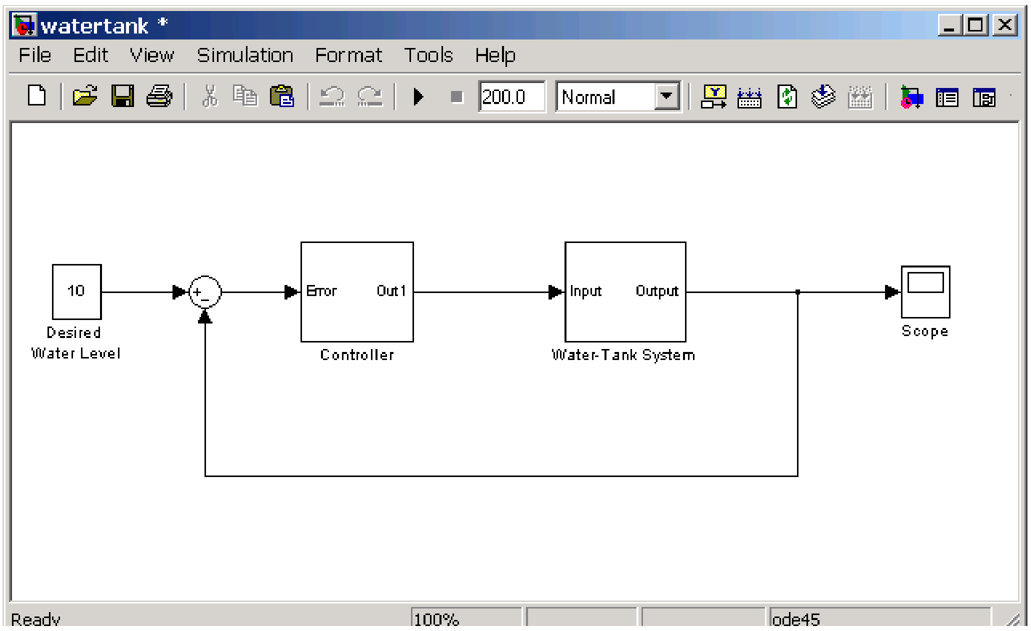


Figure 4-2: Simulink Model of the Water-Tank System

The `watertank` model consists of

- The water-tank system itself
- A Controller subsystem to control the height of water in the tank by varying the voltage applied to the pump
- A reference signal that sets the desired water level
- A Scope block that displays the height of water as a function of time

Double-click a block to view its contents. The Controller block contains a simple proportional-integral-derivative controller. The Water-Tank System block is shown in this figure.

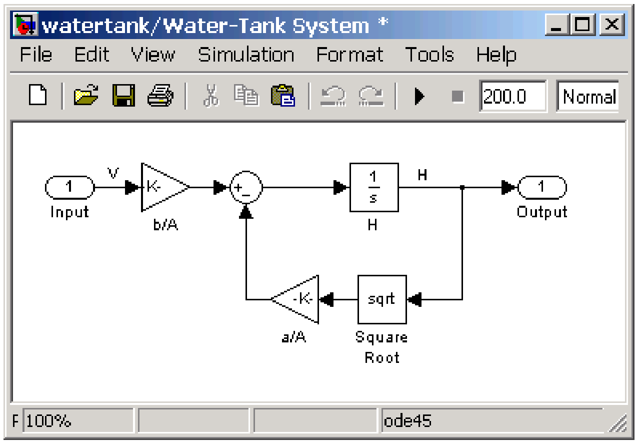

watertank/Water-Tank System *
File Edit View Simulation Format Tools Help
200.0
Normal
1
Input
V
-K-
b/A
+
−
1/s
H
H
1
Output
-K-
a/A
sqrt
Square Root
100%
ode45

Figure 4-3: Water-Tank System Block

The input to the Water-Tank System block, which is also the output of the Controller, is the voltage, V. The output is the height of water, H. The system contains just one state (within the integrator), H. Values of the parameters are given as a=2 cm$^{2.5}$/s, A=20 cm^2, b=5 cm^3/(s·V).

Configuring the Model for Linearization

Before linearizing the Simulink model of your system, configure it by

1. Choosing linearization input and output points.
2. Storing linearization points in an input/output (I/O) object.
3. Editing the I/O object, when necessary, such as when computing the open loop model.

The input and output points define the portion of your model being linearized. Setting the `OpenLoop` property of a linearization point to `'on'` allows you to compute an open loop model. Refer to "Linearization of Simulink Models" on page 2-9 for more information on linearization input and output points.

Choosing and Storing Linearization Points

In the `watertank` model, the nonlinearities are in the water-tank system itself. To linearize this portion of the model, place an input point before it and an output point after it. Information about the linearization points is stored in an input/output (I/O) object in the MATLAB workspace.

Each linearization point is associated with an outport of a block. To place an input point before the Water-Tank System block, you need to associate this input point with the outport of the Controller block.

To create an I/O object for the input point, use the `linio` function.

```
watertank_io(1)=linio('watertank/Controller',1,'in')
```

This creates an object, `watertank_io`, in the MATLAB workspace and displays the object as shown below.

```
Linearization IOs:
-------------------------
Block watertank/Controller, Port 1 is marked with the following
properties:
 - No Loop Opening
 - An Input Perturbation
```

The first input argument of the `linio` function is the name of the block that the linearization point is associated with. The second argument is the number of the outport on this block that the linearization point is associated with. These two arguments allow the linearization point to be placed on a specific signal line. The third argument is the type of linearization point. Available types are

`'in'`	input point
`'out'`	output point
`'inout'`	input point followed by output point
`'outin'`	output point followed by input point

To create a second object within `watertank_io` for an output point, use the following command.

```
watertank_io(2)=linio('watertank/Water-Tank System',1,'out')
```

This creates an I/O object for the output point that is located at the first outport of the block `watertank/Water-Tank System`. The object `watertank_io` is displayed, as shown below.

```
Linearization IOs:
--------------------------
Block watertank/Controller, Port 1 is marked with the following
properties:
 - No Loop Opening
 - An Input Perturbation

Block watertank/Water-Tank System, Port 1 is marked with the
following properties:
 - An Output Measurement
 - No Loop Opening
```

Both the input and output points are now stored in the MATLAB workspace in the I/O object `watertank_io`. To view the linearization points on the model diagram, upload the settings in `watertank_io` using the `setlinio` function.

```
setlinio('watertank',watertank_io)
```

The model diagram should now look like that in the following figure.

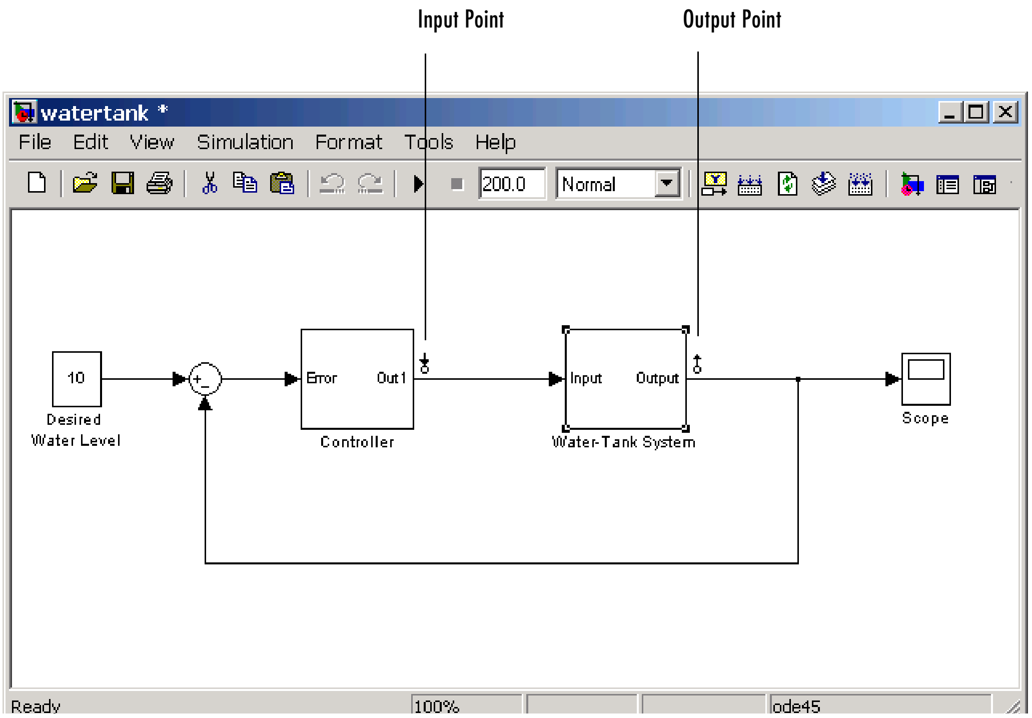


Figure 4-4: Water-Tank Model with Input and Output Points Selected

Extracting Linearization Points from a Model

An alternative way to create an I/O object is to extract the operating points from the model diagram when they have been selected using the right-click menus described in “Selecting Linearization Points” on page 3-10. The extracted linearization points are stored in an I/O object. Use the `getlinio` function to extract the linearization points in the following way.

```
watertank_io=getlinio('watertank')
```

This returns

```
    Linearization IOs:
--------------------------
```

```
Block watertank/Controller, Port 1 is marked with the following
properties:
 - No Loop Opening
 - An Input Perturbation

Block watertank/Water-Tank System, Port 1 is marked with the
following properties:
 - An Output Measurement
 - No Loop Opening
```

Editing an I/O Object

Typing the name of the I/O object at the command line returns a formatted display of key object properties. To view a list of all properties, use the `get` function. Each object within the I/O object has six properties. For example, to view the properties of the second object in `watertank_io`, type

```
get(watertank_io(2))
```

MATLAB displays

```
         Active: 'on'
          Block: 'watertank/Water-Tank System'
       OpenLoop: 'off'
     PortNumber: 1
           Type: 'out'
    Description: ''
```

You can edit this object to make any necessary changes. For example, to make this linearization point a loop opening as well, type

```
watertank_io(2).OpenLoop='on'
```

To refresh the model diagram so that it reflects any changes made to the I/O object using the functions, use the `setlinio` function.

```
setlinio('watertank', watertank_io);
```

A small × appears next to the output point in the diagram, indicating a new loop opening, as shown in this figure.

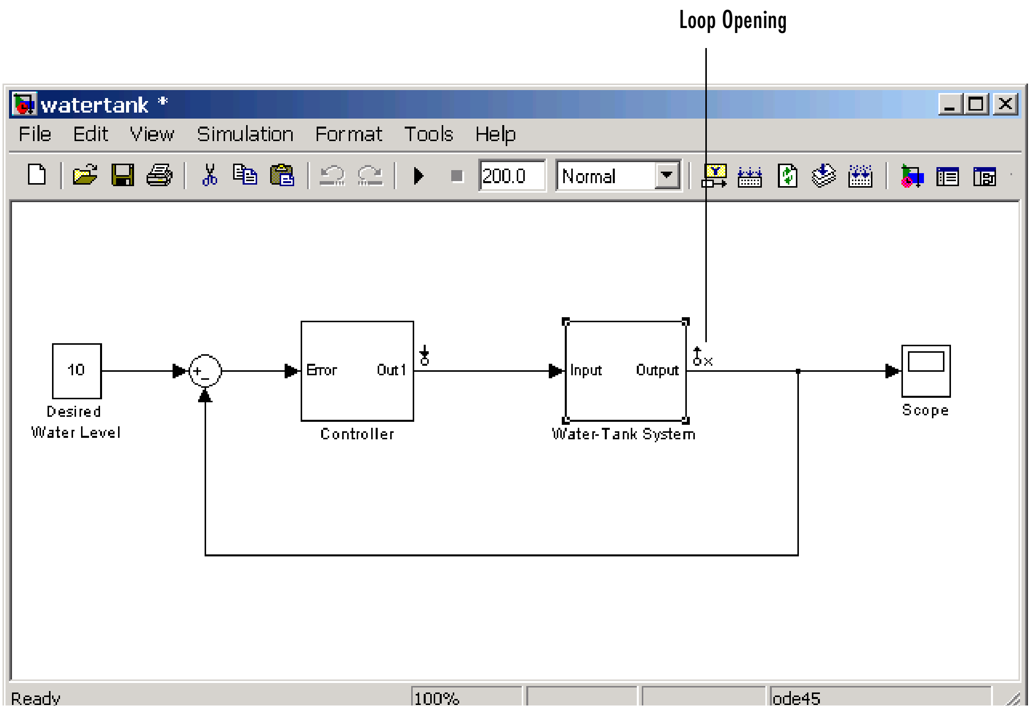


Figure 4-5: Water-tank Model with Loop Opening

You can edit the other properties of I/O objects in a similar way. For more information about each property and the possible values it can take, see the `getlinio` reference page.

Open Loop Analysis Using Functions

When you want to remove the effect of signals feeding back into the portion of the model you are linearizing, it is often convenient to insert open loop points in the model. For more information on this topic, refer to “Understanding Open Loop Analysis” on page 2-11. For methods on inserting loop openings with the Simulink Control Design GUI, refer to “Performing Open Loop Analysis” on page 3-13. An alternative method of inserting loop openings, using functions, is to edit the I/O object as described in “Editing an I/O Object” on page 4-10.

Specifying Operating Points Using Functions

Before linearizing the model, you must choose an operating point to linearize the system about. This is often a steady-state value. Refer to “Operating Points” on page 2-3 for more information on the role of operating points in linearization.

Use the Simulink Control Design functions for any of the following methods of specifying the operating point:

- You know the operating point explicitly, i.e., you know the values of all inputs and states in the model.
- You want to simulate the model and extract the operating point at a given time.
- You do not know all the input and state values, but you can characterize the operating point indirectly by specifying operating point values and constraints for specific signals and variables in the model (implicit specification).

Note The operating point consists of values for *all* the states in the model although only those states between the linearization points will be linearized. This is because the whole model contributes to the operating point values of the states/inputs/outputs of the portion of the model you are linearizing.

Specifying Completely Known Operating Points

To use functions to specify completely known operating points, first create an operating point object, and then make changes to the object values.

Creating an Operating Point Object

An operating point object contains information about your system’s states and inputs at the operating point. When you know your operating point explicitly, use the function `operpoint` to create an operating point object for your model.

For example, to create an operating point object for the water-tank model, type

```
watertank_op=operpoint('watertank')
```

MATLAB displays

```
Operating Point for the Model watertank.
(Time-Varying Components Evaluated at time t=0)

States:
----------
(1.) watertank/Controller/Integrator
      x: 0
(2.) watertank/Water-Tank System/H
      x: 0

Inputs: None
```

Within the operating point object are objects for all the states and inputs in the model. Each of these objects has a property called `x`, or `u` in the case of inputs, that gives the value of that state or input.

Changing Operating Point Values

Change the `x` and `u` properties of the operating point object to known values at the operating point. For example, to change the value of the first state to 1.26 and the second state to 10, type

```
watertank_op.States(1).x=1.26, watertank_op.States(2).x=10
```

which returns

```
Operating Point for the Model watertank.
(Time-Varying Components Evaluated at time t=0)

States:
----------
(1.) watertank/Controller/Integrator
      x: 1.26
(2.) watertank/Water-Tank System/H
      x: 10

Inputs: None
```

The operating point object, `watertank_op`, now contains the known operating point of the system.

Extracting Values from Simulation

Use Simulink Control Design to extract operating points from a simulation of your model at specified times, such as when the simulation reaches a steady state solution.

For example, to create an operating point object for the water-tank model after it has simulated for 20 time units, type

```
watertank_op=findop('watertank',20)
```

MATLAB displays the operating point at time `t=20`.

```
Operating Point for the Model watertank.
 (Time-Varying Components Evaluated at time t=20)

States:
----------
(1.) watertank/Controller/Integrator
      x: 1.25
(2.) watertank/Water-Tank System/H
      x: 9.99

Inputs: None
```

Computing Operating Points from Specifications

To determine the operating points from specifications,

1. Create an operating point specification object. See, "Creating an Operating Point Specification Object" on page 4-15.

2. Configure the object to store the specifications such as any constraints or known information about the operating point. See "Configuring the Operating Point Specification Object" on page 4-15.

3. Use the `findop` function to find the operating point values by optimization. See "Computing the Complete Operating Point" on page 4-16.

Creating an Operating Point Specification Object

When you know only some values exactly, or you know constraints on the values in the operating point, use the function `operspec` to create an operating point specification object for your model.

For example, to create an operating point specification object for the `watertank` model, type

```
watertank_spec = operspec('watertank')
```

MATLAB displays

```
Operating Specificaton for the Model watertank.
(Time-Varying Components Evaluated at time t=0)

States:
----------
(1.) watertank/Controller/Integrator
      spec:  dx = 0,  initial guess:            0
(2.) watertank/Water-Tank System/H
      spec:  dx = 0,  initial guess:            0

Inputs: None

Outputs: None
```

Configuring the Operating Point Specification Object

The operating point specification object contains objects for all the states, inputs, and outputs in the model. By typing the object's name at the command line you can see a formatted display of key object properties. Alternatively, to list all the properties for a particular object, use the `get` function. For example

```
get(watertank_spec.States(1))
```

returns

```
          Block: 'watertank/Controller/Integrator'
              x: 0
             Nx: 1
          Known: 0
    SteadyState: 1
            Min: -Inf
```

```
                Max: Inf
        Description: ''
```

Edit these properties to provide specifications for the operating point. For example:

- To set the second state to a known value (such as the desired height of water), first change `Known` to `1`.

  ```
  watertank_spec.States(2).Known=1
  ```

  Next, provide the known value.

  ```
  watertank_spec.States(2).x=10
  ```

- To find a steady-state value for the first state, set `SteadyState` to `1`.

  ```
  watertank_spec.States(1).SteadyState=1
  ```

- To provide an initial guess of `2` for this steady-state value, first make sure that `Known` is set to `0` for this state.

  ```
  watertank_spec.States(1).Known=0
  ```

  Then, provide the initial guess.

  ```
  watertank_spec.States(1).x=2
  ```

- To set a lower bound of 0 on this state,

  ```
  watertank_spec.States(1).Min=0
  ```

Optimization settings used with the `findop` function can be configured with the `linoptions` function.

Computing the Complete Operating Point

The operating point specification object, `watertank_op`, now contains specifications for the operating point. Use this information to find the complete operating point using the `findop` command. Type

```
[watertank_op,op_report]=findop('watertank',watertank_spec)
```

This returns the optimized operating point. The optimized values of the states are contained in the `x` property, or `u` property for inputs.

```
Operating Point for the Model watertank.
(Time-Varying Components Evaluated at time t=0)
```

```
States:
----------
(1.) watertank/Controller/Integrator
      x: 1.26
(2.) watertank/Water-Tank System/H
      x: 10

Inputs: None
```

The operating point search report, `op_report`, is also generated. The `x` or `u` values give the state or input values. The `dx` values give the time derivatives of each state, with desired `dx` values in parentheses. The fact that the `dx` values are both zero indicates that the operating point is at steady state.

```
Operating Point Search Report:
---------------------------------

 Operating Point Search Report for the Model watertank.
(Time-Varying Components Evaluated at time t=0)

Operating condition specifications were successully met.

States:
----------
(1.) watertank/Controller/Integrator
      x:          1.26      dx:             0 (0)
(2.) watertank/Water-Tank System/H
      x:            10      dx:             0 (0)

Inputs: None

Outputs: None
```

Alternative Method for Specifying Initial Guesses

In some cases you might want to use a previously created operating point to specify initial guesses in another operating point specification object. For example, after extracting an operating point from a simulation, as in “Extracting Values from Simulation” on page 4-14, you can use this operating point as a starting point for finding a more accurate steady state value using `findop`. You can do this with the `initopspec` function.

For example, first extract an operating point from simulation, in this case after 10 time units.

```
watertank_op2=findop('watertank',10);
```

Then create an operating point specification object.

```
watertank_spec=operspec('watertank');
```

Specify initial guesses in this object with the following command.

```
watertank_spec=initopspec(watertank_spec,watertank_op2)
```

This returns an operating point specification with the initial guess, or `x` property filled with operating point values from `watertank_op`.

```
Operating Specificaton for the Model watertank.
 (Time-Varying Components Evaluated at time t=0)

States:
----------
(1.) watertank/Controller/Integrator
      spec:  dx = 0,  initial guess:        0.872
(2.) watertank/Water-Tank System/H
      spec:  dx = 0,  initial guess:          9.7

Inputs: None

Outputs: None
```

This operating point specification can now be used with `findop` to find an optimized steady state operating point. You can access individual elements of this object using the `get` function or dot-notation as in “Configuring the Operating Point Specification Object” on page 4-15.

Adding Output Constraints to Specifications

When you want to constrain additional signals of the model, you can add an output constraint to the operating point specification object with the function `addoutputspec`.

For example, to add an output constraint to the operating point specification created in “Alternative Method for Specifying Initial Guesses” on page 4-17, use the following command.

```
watertank_spec=addoutputspec(watertank_spec,'watertank/Water-Tan
k System/Sum',1)
```

This adds a constraint on the signal between the Sum block and the integrator block, H, within the Water-Tank System block. The constraint is associated with an outport of a block, in this case outport 1 of the block 'watertank/Water-Tank System/Sum' (the preceding block).

```
Operating Specificaton for the Model watertank.
 (Time-Varying Components Evaluated at time t=0)

States:
----------
(1.) watertank/Controller/Integrator
      spec:  dx = 0,  initial guess:         0.872
(2.) watertank/Water-Tank System/H
      spec:  dx = 0,  initial guess:           9.7

Inputs: None

Outputs:
-----------
(1.) watertank/Water-Tank System/Sum
      spec:  none
```

You can edit the specifications for this output in the same way as you would for any other specifications, by changing the values of Known, y, Min, etc. There is no SteadyState option for outputs.

Using Vectors of Operating Point Values

The operating point values are stored within the operating point object. However, sometimes it is useful to work with *vectors* of operating point values, rather than the objects. To do this you can use the functions getxu and setxu.

For example, to extract a vector of operating point values from the operating point object, watertank_op, created in “Computing the Complete Operating Point” on page 4-16, use the following command.

```
[x,u]=getxu(watertank_op)
```

This extracts vectors of states, x, and inputs, u, from the operating point object, as shown below.

```
x =
   10.0000
    1.2649

u =
     []
```

The ordering of states in these objects is the same as the ordering used by Simulink but not necessarily the same as the order the states appear in the operating point object.

To set operating point values in an operating point object using a vector of known values, you can use the following command.

```
new_op=setxu(watertank_op2,x,u)
```

This sets the values in x and u as the values of states and inputs in the operating point object. If x and u are the same as those extracted above, then the operating point object will not change. To set the operating point values to 1, first create a vector of ones.

```
x=[2;1]
x =
     1
     1
```

Then assign these numbers to the state values in watertank_op.

```
new_op=setxu(watertank_op2,x,u)
```

This returns

```
Operating Point for the Model watertank.
(Time-Varying Components Evaluated at time t=0)

States:
----------
(1.) watertank/Controller/Integrator
      x: 1
```

```
(2.) watertank/Water-Tank System/H
      x: 2

Inputs: None
```

Note that the ordering of numbers in the vector x is different from the ordering in the operating point object.

Linearizing the Model

After creating an I/O object and determining the operating point, you are ready to linearize the system, using the `linearize` command. For example:

```
watertank_lin=linearize('watertank',watertank_op,watertank_io)
```

MATLAB will return the matrices `a`, `b`, `c`, and `d` of a linear, time-invariant, state-space model that approximates your nonlinear system in a region around the operating point.

```
a =
                  watertank/Wa
   watertank/Wa       -0.01581

b =
                  watertank/Co
   watertank/Wa          0.25

c =
                  watertank/Wa
   watertank/Wa             1

d =
                  watertank/Co
   watertank/Wa             0

Continuous-time model.
```

To change the linearization options, use the `linoptions` function before running the linearization. For example, to change the sample time for the linearization model to be `1` instead of continuous, use the following command.

```
linopt=linoptions('SampleTime',1);
```

Then, run the linearization with these options.

```
watertank_lin2=linearize('watertank',watertank_op,watertank_io,l
inopt)
```

This returns the discrete-time model shown below.

```
a =
                 watertank/Wa
   watertank/Wa        0.9843


b =
                 watertank/Co
   watertank/Wa        0 .248


c =
                 watertank/Wa
   watertank/Wa             1


d =
                 watertank/Co
   watertank/Wa             0

Sampling time: 1
Discrete-time model.
```

Linearizing Discrete-Time and Multi-Rate Models

The linearization method is the same for models containing discrete-time states or several different sampling rates. However, you can choose to adjust the `SampleTime` parameter with the `linoptions` function as shown in the previous section. By default this parameter is set to `-1`, in which case Simulink Control Design will find the slowest sample rate in the model to use for the sample rate of the linearized model. To create a linearized model with different sample time, specify a new parameter value before linearizing the model. A value of `0` will give a continuous-time model.

Analyzing the Results

Analyze the linearized model at the MATLAB prompt using functions from the Control System Toolbox, or display it in the LTI Viewer. Alternatively, incorporate the results into a block in a Simulink model. For methods on simulating the linearized model for comparison with the original model, refer to “Comparing the Linearized and Original Models” on page 5-2.

Using the LTI Viewer

To send your linearized model to the LTI Viewer for display, type

```
ltiview(watertank_lin)
```

The LTI Viewer opens, by default, with a step response of the linearized system, as shown in the following figure.

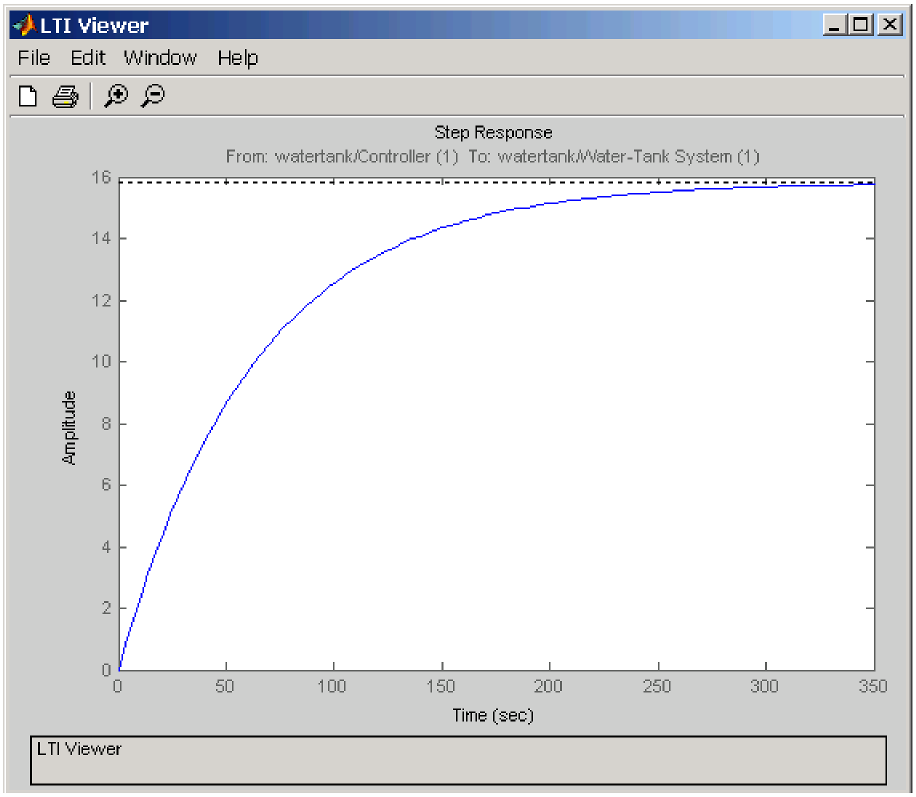


Figure 4-6: LTI Viewer Displaying a Step Response of the Linearized Model

You can use standard LTI Viewer features to display your results. For example, change the plot type by right-clicking anywhere in the plot area and choosing from the **Plot Types** menu. To add characteristics such as settling time or peak response to your plot, right-click anywhere in the plot area and choose from the **Characteristics** menu. Add data markers by clicking the point you want to mark.

You can display up to six plots in the LTI Viewer window. To change the number of plots, select **Edit -> Plot Configurations**, choose a configuration in the **Plot Configurations** window, and then click **OK**.

For more information on the LTI Viewer, refer to the Control System Toolbox documentation.

Saving Your Work

To save your linearized model for later analysis, use the `save` command. For example, to save the linearized model, operating points, and I/O object of the `watertank` model, type

```
save watertank_project watertank_lin watertank_op watertank_io
```

This creates a file named `watertank_project.mat` in the current directory. To reload this file, use the `load` function.

```
load watertank_project
```

Restoring Linearization I/O Settings

To save linearization I/O settings for use in a later session, use the `save` function. You can then restore the settings by loading them with the `load` function and using the `setlinio` function to upload them to the model diagram. For more information on setio see the function reference page for `setlinio`.

Alternatively, you can use the reloaded I/O settings object with the `linearize` function without uploading it to the model diagram.

Index